



**Universitat Autònoma
de Barcelona**

iSustainable

Memòria del projecte
d'Enginyeria Tècnica en
Informàtica de Gestió
realitzat per
Roger Domènech Aguilera
i dirigit per
Daniel Franco Puntos
Lluís Rodríguez Barquero

Escola Universitària d'Informàtica
Sabadell, Juny de 2009

El sotasignant, Daniel Franco Punes,
professor/a de l'Escola Universitària d'Informàtica de la UAB,

CERTIFICA:

Que el treball al que correspon la present memòria
ha estat realitzat sota la seva direcció
per en Roger Domènech Aguilera.

I per a que consti firma la present.
Sabadell, Juny de 2009.

Signat: Daniel Franco Punes

El sotasignant, Lluís Rodríguez Barquero,
de SOLVAY IBERICA S.L.,

CERTIFICA:

Que el treball al que correspon la present memòria
ha estat realitzat sota la seva supervisió
per en Roger Domènech Aguilera.

I per a que consti firma la present.
Sabadell, Juny de 2009.

Signat: Lluís Rodríguez Barquero.

RESUM DE LA MEMÒRIA

iSustainable és un projecte relacionat amb la tecnologia iPhone d'Apple. L'iPhone és un terminal mòbil amb absència de botons i amb una gran pantalla que ocupa pràcticament la totalitat d'una de les seves cares. Aquest aparell simula el funcionament d'un ordinador amb el seu sistema operatiu, connexió a Internet, gestió de fitxers, instal·lació i ús d'aplicacions descarregades via web, etc... El projecte iSustainable conjuga el desenvolupament i investigació en aquesta tecnologia amb un projecte ja existent dins de la companyia com és el Solvay Sustainable. El projecte Solvay Sustainable és un projecte de llarg recorregut per afrontar els reptes de desenvolupament sostenible i creixement econòmic que marcaran el futur de la companyia en les pròximes dècades.

El projecte es realitza en el marc de l'empresa Solvay, la qual és una multinacional belga dels sectors químic, plàstic i farmacèutic. Fundada el 1863 per Ernest Solvay, químic belga, i amb més de 100 d'anys d'implantació a Espanya. És realitza un estudi de viabilitat del projecte per a decidir la conveniència o no de la realització del mateix, malgrat que no es poden quantificar exactament els beneficis, degut a que són uns beneficis a llarg termini es decideix tirar endavant el projecte, ja que un dels principals objectius del projecte és l'aprenentatge i desenvolupament en aquesta tecnologia.

El projecte iSustainable esta enfocat a càrrecs directius de la companyia, els quals necessitin consultar les dades proporcionades per l'aplicació en tot moment. Aquesta aplicació s'actualitzarà mitjançant la descàrrega d'una BBDD col·locada en un servidor. Aquesta BBDD l'actualitzarà alguna persona pertanyent al personal administratiu de l'empresa mitjançant una aplicació web inclosa també en el projecte iSustainable. Aquesta aplicació serà posada en producció en un dels servidors interns de la companyia i realitzarà l'actualització en un servidor extern per a la seva posterior descàrrega a l'iPhone.

Posteriorment a la fase d'aprenentatge es realitza un disseny primari del conjunt, creant després el disseny de la aplicació iPhone i posteriorment la BBDD i la aplicació d'actualització seguint en tots els casos els requeriments de l'empresa. L'entorn de desenvolupament de tot el programari ha estat molt condicionat a la tecnologia que es vol emprar, sobretot en un principi en l'aplicació iPhone i la BBDD, la aplicació web d'actualització es pot implementar de forma més lliure elegint el programari que es consideri més adient. La programació d'aplicacions iPhone està condicionada al llenguatge Objective-C i a tot l'entorn de programació d'Apple: Xcode, Interface Builder, iPhone Simulator, etc... El desenvolupament de la BBDD també està lligat al terminal, degut a que aquest només funciona amb el format SQLite de BBDD, en aquest sentit s'ha emprat l'eina pel navegador Mozilla Firefox SQLite Manager. Per a l'aplicació d'actualització s'ha utilitzat l'eina de programari lliure Turbogears, la qual utilitza el llenguatge de programació Python.

La fase d'implementació ha seguit la mateixa línia que la fase de disseny, començant per l'aplicació iPhone i després evolucionant cap a la BBDD i la aplicació d'actualització. En aquestes fases s'han solucionat els diferents problemes que han aparegut, realitzant els canvis necessaris en la idea original, però assolint els diferents objectius marcats. També s'han realitzat les diferents fases de proves de cada aplicació fins aconseguir un rendiment òptim de cadascuna d'elles. Un cop assolits tots els punts anteriors s'ha prosseguit a extreure les conclusions del projecte, pensar nous àmbits en els que aplicar aquesta tecnologia i realitzar la documentació pertinent.

TAULA DE CONTINGUTS

1. Introducció.....	9
1.1 Motivacions generals.....	9
1.2 Objectius.....	10
1.3 Motivacions personals.....	10
1.4 Usuaris	11
1.5 Organització del projecte	11
1.6 Estructura de la memòria	12
2. L'empresa	14
2.1 El Grup Solvay	14
2.2 Sectors d'activitat	15
2.2.1 Sector Químic	15
2.2.2 Sector Plàstics	16
2.2.3 Sector Farmacèutic	16
2.3 El Grup Solvay a Espanya	16
2.4 SIS (Solvay Information Services).....	16
3. Definició del projecte	18
3.1 El terminal iPhone	18
3.2 Necessitats de l'empresa.....	20
3.2.1 Solvay Sustainable	21
3.3 El projecte.....	23
3.3.1 La idea.....	24
3.4 Resum	24
4. Estudi de viabilitat	25
4.1 Introducció.....	25
4.1.1 Avantatges	25
4.1.2 Inconvenients	25
4.2 Objecte	26
4.2.1 Situació a tractar.....	26
4.2.2 Perfil de l'usuari	26
4.2.3 Objectius.....	27
4.3 Descripció del sistema	28
4.3.1 Descripció	28
4.3.2 Recursos	29
4.3.3 Avaluació de Riscos.....	30
4.3.4 Organització del projecte	30
4.4 Planificació del projecte	31
4.5 Conclusions.....	32
5. Anàlisi i disseny.....	33
5.1 Disseny troncal	33
5.2 Disseny de la aplicació iPhone.....	35
5.3 Disseny de la BBDD	37
5.4 Disseny de la aplicació d'actualització.....	39
5.5 Resum	40

6. Entorn de desenvolupament	41
6.1 Objective-c	41
6.1.1 Història	41
6.1.2 Característiques	42
6.1.3 Objective-c 2.0	43
6.2 Xcode	46
6.2.1 Característiques	47
6.3 Interface builder	48
6.3.1 Funcionament	48
6.3.2 Característiques	49
6.4 SQLite	50
6.4.1 Característiques principals	51
6.5 Turbogears	52
6.5.1 Patró model-view-controller	53
6.5.2 Mochikit	53
6.5.3 Kid	54
6.5.4 Cherrypy	55
6.5.5 Python	55
6.5.6 SQLAlchemy	57
7. Implementació	59
7.1 iPhone	59
7.1.1 Descripció del codi	60
7.1.2 Exemples d'algoritmes	63
7.2 Aplicació d'actualització	64
7.2.1 Controllers	65
7.2.2 Forms	67
7.2.3 devdata.sqlite	67
7.2.4 Templates	68
7.2.5 css, images i files	69
7.3 Fase de proves i distribució	69
7.3.1 Mètode Caixa Negra	69
7.3.2 Mètode Caixa Blanca	70
7.3.3 Proves iPhone	70
7.2.4 Proves aplicació d'actualització	72
8. Conclusions	73
8.1 Acompliment dels objectius	73
8.2 Altres aspectes a destacar	74
8.3 Línies Obertes	75
8.4 Valoració personal i agraïments	76
9. Bibliografia	77
Anex 1: Manual d'usuari iSustainable	79
1. Requisits mínims	79
2. Instal·lació	79
Pas 1: Contactar amb l'administrador	79
Pas 2: Descarregar l'aplicació	80
Pas 3: Instal·lació	80

3. Funcionament.....	81
Anex 2: Manual d'usuari Aplicació d'actualització	86
1. Requisits mínims.....	86
2. Instal·lació.....	86
3. Funcionament.....	86
Pas 1: Identificació.....	86
Pas 2: Menú principal	87
Pas 3: Edit Database	88
Pas 4: Seleccionar cel·la de la Matriu Solvay Sustainable.....	89
Pas 5: Modificar el contingut de la cel·la	90
Pas 6: Import database, Download zip file i Download profile.....	92

1. INTRODUCCIÓ

1.1 MOTIVACIONS GENERALS

La idea de la realització d'aquest projecte es fonamenta en la voluntat constant d'innovació i aprenentatge del departament d'informàtica de l'empresa Solvay. Dins d'aquest, s'intenta fomentar el coneixement de noves tecnologies tant en el món de la informàtica com en el de les telecomunicacions, s'estudien les seves possibles aplicacions empresarials i es valora la conveniència de la seva implementació en el marc de l'empresa.

En aquest context es contempla la revolució que s'està produint en el sector de la telefonia mòbil i es considera la possibilitat de realitzar un estudi de les alternatives que van apareixen en el mercat. Es planteja la possibilitat de realitzar un conveni de col·laboració universitat-empresa, per fer l'estudi. En aquest es decideix apostar per la plataforma iPhone d'Apple.

Aquesta decisió es pren com a resultat de l'estudi previ, el qual ens proporciona una sèrie d'arguments a favor i en contra d'aquesta decisió. S'avalua també la solució proposada per l'empresa Google i només es consideren aquestes solucions, ja que solucions anteriors com Debian o Windows Mobile es desestimen prèviament. A continuació detallarem els diferents avantatges i inconvenients d'ambdues solucions:

- **Solució Apple:**
 - **Avantatges:**
 - Maduresa: producte en funcionament en el mercat USA durant l'últim any.
 - Servei tècnic en el desenvolupament per la compra d'una llicència de pagament.
 - Confiança en el fabricant: gran experiència i reputació en el desenvolupament d'altres productes informàtics i electrònics.
 - Aparició imminent a Europa i Espanya.
 - Èxit: gran acceptació del producte entre la societat.
 - Sistema de distribució consolidat iTunes, App Store per a aplicacions iPhone. Plataforma exclusiva per a la distribució d'aplicacions iPhone.
 - **Inconvenients:**
 - Major inversió per la compra d'un terminal de la marca Apple i per la obligatorietat d'adquirir una llicència de pagament per a distribuir la aplicació.
 - Llenguatge específic de desenvolupament: obligatorietat d'aprendre un llenguatge diferent amb les seves particularitats.
 - Dificultat d'integració: estudi d'integració de tecnologia Apple en entorn Microsoft.
- **Solució Google:**
 - **Avantatges:**
 - Preu: absència de maquinari propi i llicència.
 - Llenguatge de programació Java.

- Inconvenients:
 - Absència d'un terminal propi.
 - Producte de recent aparició.
 - Incertesa del grau d'implantació en el mercat.
 - Adaptació a cada terminal.

Posteriorment es valora positivament el treball d'investigació i estudi realitzat durant els mesos d'estiu i les possibilitats que ofereix la tecnologia. Es decideix ampliar el conveni de col·laboració per a realitzar un projecte de final de carrera. El següent projecte és el resultat d'aquest conveni.

1.2 OBJECTIUS

Un cop triada la plataforma que considerem més adient ens marquem una sèrie d'objectius a accomplir cronològicament:

- Conèixer suficientment l'entorn MAC OS per a poder treballar-hi amb la comoditat i rapidesa necessàries per a desenvolupar-hi.
- Descobrir les eines de desenvolupament en entorn MAC i iPhone, conèixer les seves particularitats i les diferents prestacions que proporcionen al desenvolupador.
- Aprendre el llenguatge de programació Objective-C: conèixer en profunditat les seves principals característiques per a poder desenvolupar aplicacions de forma ràpida i eficient.
- Analitzar les necessitats de l'empresa i com les solucions que aporta la tecnologia d'Apple ens poden ajudar a resoldre-les, sobretot en el cas de l'iPhone i les seves característiques.
- Dissenyar un prototip d'aplicació utilitzant els coneixements adquirits i tenint en compte els requeriments de la companyia.
- Enriquir aquest prototip amb múltiples millores successives que aportin noves solucions, enriqueixin la aplicació o en facilitin el seu ús.
- Desenvolupar altres aplicacions paral·leles per a cobrir necessitats menys rellevants en el context de l'empresa.
- Establir la tecnologia i els procediments necessaris per a futurs desenvolupaments en aquesta plataforma en el si de la companyia, els qual puguin prendre aquest desenvolupament com un punt de partida cap a noves aplicacions Apple o iPhone.

1.3 MOTIVACIONS PERSONALS

La realització d'un projecte informàtic requereix d'un gran esforç, per tant el grau de motivació de l'alumne ha de ser molt alt per poder accomplir els objectius marcats. Però en aquest projecte en concret la voluntat de superació personal, l'esforç intel·lectual per aprendre, descobrir i innovar i l'esperit crític han de prevaldre per a poder desenvolupar des de zero en una tecnologia nova i completament desconeguda per la majoria on cada pas és un desafiament i només l'esperança d'un èxit final i la

culminació d'una carrera universitària tant fermament desitjada fan perseverar l'alumne en la direcció correcta per a concretar un procés llarg i difícil.

1.4 USUARIS

A priori els usuaris als que es dirigeix aquest projecte són treballadors de la empresa Solvay (la qual introduïrem en el següent capítol). El perfil de l'usuari és alt, està enfocat sobretot a directius que viatgen molt i a més a més necessiten estar permanentment connectats a la companyia i consultar les dades en qualsevol lloc i de la manera més còmoda possible. Per això es volen introduir terminals cada cop més petits i així afavorir la comoditat en els viatges i també en el treball dia a dia, però sense perdre qualitat en el servei. En un futur es poden desenvolupar altres projectes per a usuaris de més baix nivell però que la seva especialització els permet treure profit d'un terminal amb la potència i característiques del iPhone.

1.5 ORGANITZACIÓ DEL PROJECTE

S'utilitzarà una metodologia de prototip, ja que es tracta d'una tecnologia completament nova i desconeguda i s'implementarà en un sistema nou a l'empresa. La metodologia de desenvolupament de prototip s'implementa quan no és possible determinar tots els requeriments de l'usuari, per això inclou el desenvolupament interactiu o de continua evolució, on l'usuari o client participa de forma directa en el procés de desenvolupament.

Aquest mètode s'implementa també en casos en els que els encarregats del desenvolupament tenen poca experiència o informació i/o els costos i riscos de que es cometi un error poden ser alts. Així mateix aquest sistema resulta útil per a provar la facilitat del sistema i identificar els requisits de l'usuari, avaluar el disseny d'un sistema o examinar l'ús d'una aplicació.

Aquest mètode es compon de 5 etapes bàsiques, com també es pot veure en la figura N° 1:

1. **Identificació dels requeriments coneguts prèviament:** en aquesta etapa els analistes i els usuaris han de treballar plegats per a identificar els màxims requeriments possibles.
2. **Desenvolupament d'un model de treball:** el desenvolupament d'un pla general de treball facilita el inici del procés i permet saber als usuaris què s'espera d'ells i del procés de desenvolupament.
3. **Utilització del prototip:** és responsabilitat de l'usuari treballar amb el prototip i avaluar les seves característiques. L'experiència del sistema en condicions reals permet obtenir la familiaritat indispensable per a identificar els canvis i millores necessaris i les característiques inadequades.
4. **Revisió del prototip:** durant l'avaluació, els analistes han de ser capaços de capturar la informació respecte al que els hi agrada o no als usuaris.
5. **Repetició del procés:** el procés anterior es realitza tantes vegades com sigui necessari fins que l'usuari i l'analista estan d'acord en que el sistema ha evolucionat de forma suficient.

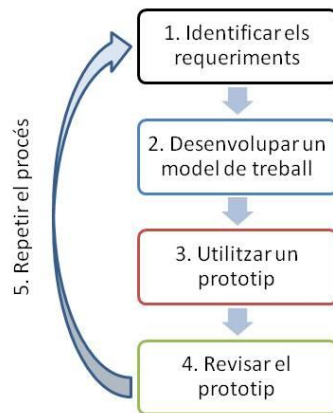


Figura nº 1.

En el projecte s'han adaptat les etapes abans descrites als requeriments concrets d'aquest projecte:

1. Aprenentatge de la tecnologia, llenguatge de programació i funcionament de les diferents eines de desenvolupament en iPhone.
2. Entrevista amb els responsables del projecte per a definir el tipus d'aplicació a realitzar i el seus requeriments funcionals.
3. Estudi detallat per adaptar els requeriments a la tecnologia.
4. Disseny de l'aplicació iPhone i de la BBDD.
5. Implementació del codi i creació de la BBDD.
6. Fase de proves de l'aplicació iPhone.
7. Estudi de les diferents alternatives per a realitzar el programari que actualitzarà la BBDD.
8. Fase d'aprenentatge de Turbogears i llenguatge Python.
9. Disseny de l'aplicació per a actualitzar la BBDD.
10. Implementació del codi.
11. Fase de proves.
12. Elaboració de documentació i manuals d'ús.

1.6 ESTRUCTURA DE LA MEMÒRIA

La memòria està estructurada de forma evolutiva, construïda perquè el lector vagi adquirint els coneixements paulatinament i de forma continuada, introduint en cada capítol nous conceptes que seran útils en el següent. Per començar amb la introducció de la memòria i després la companyia i el departament on s'ha desenvolupat el projecte. Posteriorment explicar el projecte amb el qual es col·labora dins de l'empresa i l'estudi de viabilitat; per continuar amb el disseny, l'entorn de

desenvolupament i finalment la implementació i la fase de proves del mateix. Per a acabar amb les conclusions extretes del projecte.

Capítol 3, L'empresa: Introdueix l'empresa, la posició de Solvay al món i les seves diferents branques de producció. També s'explica la història de Solvay a Espanya i les plantes existents en aquests moments. Finalment s'introdueix el departament on s'ha desenvolupat el projecte.

Capítol 4, Definició del projecte: Introdueix el lector en la tecnologia iPhone, també es comenten les necessitats de l'empresa i el Projecte Solvay Sustainable. Finalment s'explica la idea d'afegir el projecte de desenvolupament en iPhone al Solvay Sustainable.

Capítol 5, Estudi de viabilitat: S'analitzen els avantatges i inconvenients de la realització del projecte. Quantificant els seus costos i valorant els beneficis.

Capítol 6, Anàlisi i disseny: Es realitza el disseny general del projecte, posteriorment s'endinsa en el disseny de cada mòdul del projecte per separat. Començant per la aplicació iPhone, la base de dades i finalment la aplicació d'actualització.

Capítol 7, Entorn de desenvolupament: S'han analitzat les diferents eines de desenvolupament emprades per a la realització del projecte. Exposant les seves virtuts, defectes i perquè s'ha pres la decisió d'emprar-les.

Capítol 8, Implementació: Es comenta la realització del projecte, amb els problemes que han aparegut i com s'han solucionat. També s'explica les fases de proves dels diferents mòduls del projecte.

Capítol 9, Conclusions: Es repassa la consecució dels diferents objectius marcats a l'inici del projecte. Posteriorment es comenten els diferents punts destacats i a continuació es comenten les línies obertes després de la seva realització i com aquestes poden evolucionar. Finalment es comenta de forma personal el que ha significat la realització d'aquest projecte.

2. L'EMPRESA

El projecte ha estat realitzat al Grup Solvay¹, el qual és un grup internacional amb seu a Brussel·les. Situat a 50 països en tot el món inicià les seves activitats amb el procés de producció del carbonat sòdic, descobert per Ernest Solvay l'any 1.861. Actualment, és una empresa situada entre les primeres del món en els seus sectors d'activitat: químic, plàstic i farmacèutic. A continuació realitzarem un breu repàs a l'història i situació de la companyia, i presentarem el departament on s'ha realitzat el projecte, la SIS (Solvay Information Services). D'aquesta forma, ens introduïm en el marc on s'ha desenvolupat el projecte.

2.1 EL GRUP SOLVAY



El grup va ser fundat l'any 1.863 per explotar el procés de fabricació del carbonat sòdic a partir de l'amoníac, procés descobert per Ernest Solvay. Actualment és una empresa química i farmacèutica mundial, especialitzada en els sectors d'activitat: farmacèutic, químic, plàstic i de transformació. En els seus 400 centres repartits en 50 països de tot el món hi treballen més de 33.000 persones, 2.500 de les quals es dediquen a la investigació.

El seu volum anual de negocis supera els 7.800 milions d'euros.

La política del grup es basa en concentrar els esforços en un nombre limitat d'activitats que contribueixen al benestar de la comunitat i per a les quals es posseeix un gran coneixement tecnològic i comercial. Els productes de Solvay són presents de forma constant en la nostra vida diària; en coses tant importants com l'alimentació, la roba, la llar, els transports, les comunicacions, etc... Solvay aposta pel "compromís de progrés", com ho demostra la contrastada seguretat de les seves instal·lacions. Un equip de 450 persones a tot el món té responsabilitat directa en aquesta tasca, així com en la millora de la salut personal i la protecció del medi ambient.



■ Presence

Figura nº 2.

A Espanya, Solvay hi és present des de 1904, any en que s'inicià la construcció de la fàbrica de Torrelavega. A l'actualitat, el grup Solvay compta amb 11 centres al nostre país, on hi treballen més de 2.500 persones. La xifra de vendes supera els 640 milions d'euros, un 8% aproximadament del total a nivell mundial. Les dades de la figura 3 mostren la posició en el mercat del grup Solvay com a fabricant en cadascun dels seus productes, tant a Europa com a la resta del Món.

¹ (Solvay SA, Solvay Iberica, 2009)

El Grup Solvay al món		
	Europa	Món
Sector químic		
Carbonat sòdic	1	1
Peròxid d'hidrogen	1	1
Persals	1	1
Carbonats de bari i estronci	1	1
Sosa Càustica	1	3
Sal	1	4
Sector plàstics		
PVC	2	4
Polietilè d'alta densitat (PEHD)	1	3
Sector transformació		
Dipòsits de gasolina de PEHD	1	1
Làmines d'estanqueïtat	1	-
Tubs	3	6
Sector farmacèutic		
Antiespasmòdics	1	1
Digestius	1	1
Laxants	1	3
Antidepressius	13	9
Tractaments hormonals		
<i>Femenins</i>	5	2
<i>Masculins</i>	-	1

Figura nº 3.

2.2 SECTORS D'ACTIVITAT

La política del Grup ha consistit sempre en centrar els seus esforços en aquells sectors i productes on compta amb sobrada experiència i "know how". La diversificació de les activitats de la companyia els ha portat a estar presents en tres sectors diferents amb una única estratègia industrial i comercial.

2.2.1 SECTOR QUÍMIC

És un dels líders mundials d'aquest sector tant en essencials (carbonat sòdic, carbonats de bari i estronci, sosa càustica i peròxid d'hidrogen) com en especialitats (productes fluorats, carbonats de bari i estronci d'alta puresa, caprolactonats i bicarbonat refinat, etc...). A Espanya, el grup desenvolupa la seva activitat a través de les empreses Solvay Química, S.L., Solvay Internox, S.A, Solvay Flúor Ibérica, S.A i Solvay Minerales, S.A i dels centres de producció de Torrelavega, Tarragona i Escúzar.

2.2.2 SECTOR PLÀSTICS

En aquest sector el grup té un alt grau de diversificació i és present en mercats en expansió als cinc continents. L' utilització de tecnologia punta i el respecte pel medi ambient ha contribuït a posicionar les empreses del Grup a Espanya. Hispavic Ibérica, S.L, Solvay Benvic Ibérica, S.A i els seus respectius centres de producció a Martorell i Montornès del Vallès, estan al capdavant de la indústria espanyola de resines i compostos de PVC. La presencia de Solvay en el camp de les matèries plàstiques es complementa amb la seva participació en la transformació d'aquests materials, utilitzant les tècniques d'injecció, calandrat, extrusió, ventat, etc. Solvay és el primer proveïdor mundial de dipòsits de carburant en material plàstic. A Espanya, el Grup participa de forma destacada en aquest sector a través de la seva filial Inergy Automotive Systems Valladolid, S.L.

2.2.3 SECTOR FARMACÈUTIC

En la dècada dels anys 60, el grup Solvay s'inicia en el camp de la farmàcia integrant en la seva empresa importants societats en aquest sector. Actualment Solvay s'ha convertit en un dels 50 principals grups farmacèutics del món. A Espanya, el Grup Solvay Pharma produeix i comercialitza prop de 40 especialitats farmacèutiques per el tractament de malalties cardiovasculars, gastrointestinals, del aparell locomotor, del sistema nerviós central i de teràpia hormonal substitutiva; les activitats de producció de les quals es concentren a la fàbrica de Parets del Vallès (Barcelona).

2.3 EL GRUP SOLVAY A ESPANYA

La presència del Grup Solvay a Espanya² es remunta al 1904 amb la construcció d'una fàbrica per a la producció de sosa càustica i carbonat sòdic a Torrelavega (Cantàbria). Amb l'adquisició el 1920 de les mines de potassa de Súria (Barcelona), Solvay inicia les seves activitats a Catalunya. La presència a Espanya es consolida el 1947 amb l'establiment de la Direcció General a Barcelona. El 1972 entra en funcionament la fàbrica de PVC de Martorell. La qual, juntament amb la de Torrelavega constitueixen els pilars bàsics de la companyia a Espanya en quan a la producció química.

2.4 SIS (SOLVAY INFORMATION SERVICES)



El departament Solvay Information Services (SIS) és proveïdor intern a Solvay de tot el que refereix a les tecnologies de la informació; treballa com una "Strategic Business Unit" o SBU (Divisió unitària dins d'una empresa que treballa de manera independentment de les altres) dins del marc corporatiu de Solvay. La SIS s'encarrega de la aportació de solucions

² (Solina, E. 1999)

mitjançant canals de comunicació, coneixement del personal i potenciació dels serveis interns en concordança amb les pròpies decisions del Grup. És una SBU a nivell internacional, dins de la qual es troba el departament que s'encarrega d'Espanya i Portugal amb el nom de SIS Ibèrica. Les activitats que es realitzen dins d'aquesta organització tenen, com a objectiu principal, donar suport a les altres àrees de la corporació. Dins d'aquest conjunt de tasques podem trobar funcions com la del manteniment de sistemes informàtics, suport SAP, resolució d'incidències, desenvolupament de portals, intranets, etc. A més, la SIS també té recursos destinats al desenvolupament de solucions, tant per el propi grup Solvay com per a externs. Això fa que treballi amb certa independència del grup i tingui els seus propis clients.

3. DEFINICIÓ DEL PROJECTE

El projecte s'ha desenvolupat dins el marc del departament de la SIS. Aquest departament aposta per la innovació constant en tots els àmbits de la informàtica i les telecomunicacions dins de la companyia. El seu objectiu és anar un pas més endavant que la resta per a proveir de solucions que millorin els processos, les comunicacions, l'àmbit de treball de les persones, etc... En aquest context s'ha observat la revolució que s'està produint en el sector de la telefonia mòbil, degut a l'aparició d'un nou concepte de terminal, l'iPhone.

En el marc d'aquesta situació el departament es decideix a investigar en aquest sentit. Per a poder conèixer i aprofundir en la tecnologia, i així avaluar la conveniència o no d'integrar-ho en el conjunt de productes i solucions que ofereix el departament a la resta d'integrants de la companyia. I si és interessant desenvolupar programari propi per aquest terminal o deixar-lo només com un dels terminals mòbils amb els que treballa.

Partint d'una petita investigació prèvia, realitzada per membres del mateix departament; el més de juliol es contracta un estudiant, en aquest cas jo, en el marc del programa de pràctiques que realitza el Grup Solvay cada estiu. Durant els mesos de juliol, agost i primera quinzena de setembre s'enceta una investigació per avaluar les diferents capacitats que ofereix aquest terminal i decidir si és viable la realització d'un desenvolupament propi. Aquest estudiant ha de portar el gruix de la investigació i és l'encarregat de l'aprenentatge del codi Objective-C (que veurem més endavant) i de les diferents eines de desenvolupament que proporciona el SDK d'Apple per a iPhone.

Finalitzat aquest període es decideix de continuar el projecte; el qual es considera de suficient envergadura i complexitat com per a realitzar-ne un projecte de final de carrera. Per aquest projecte són necessaris els coneixements adquirits en diferents assignatures cursades durant els estudis, com metodologia de la programació I i II, estructura de dades, bases de dades, xarxes i d'altres.

Per a continuar amb el projecte i assolir els diferents objectius es decideix conjugar amb la iniciativa de la companyia Solvay Sustainable. Solvay Sustainable neix amb la vocació de preparar l'empresa per afrontar els diferents reptes que haurà d'encarar en aquest inici de segle i que marcaran el seu esdevenidor en els pròxims anys en temes tant importants com el medi ambient, un accés barat i segur a fonts d'energia, etc... Aquest, l'introduïrem més extensament en aquest mateix capítol.

3.1 EL TERMINAL IPHONE

Aquest terminal, l'iPhone del fabricant Apple, és el primer en sortir al mercat amb aquestes característiques i el qual revoluciona; proporciona un entorn completament nou per a la interacció amb ell, destacant l'absència pràcticament completa de botons de tipus físic i aportant una sèrie de solucions innovadores.

El primer cop que tens contacte amb un terminal d'aquestes característiques (com podem observar a la figura 4) et sorprèn l'absència total de botons, tal i com estàvem acostumats fins ara; la seva gran qualitat d'imatge i la interacció amb la pantalla tàctil.



Figura n° 4.

Aprofundint una mica més en les seves característiques descobrirem la seva potent connexió a la xarxa 3G; capacitat GPS; compatibilitat amb aplicacions per a empreses com Microsoft Exchange; la possibilitat de descàrrega de innombrables aplicacions mitjançant "App Store" d'Apple; reproductor de música, vídeo i imatges; accés total a Internet amb correu electrònic HTML enriquit; connexió Wi-fi i bluetooth; acceleròmetre; pantalla tàctil i diferents sensors de llum i proximitat; etc...

A totes aquestes característiques citades en el paràgraf anterior s'hi ha d'afegir potser, la més important, la capacitat de realitzar programari específic tant per a ser comercialitzat a través del App Store o per a ús intern d'una empresa. Mitjançant el qual les seves capacitats es veuen augmentades exponencialment i on amb una bona dosi de creativitat es poden crear aplicacions realment útils per a qualsevol tipus d'empresa o usuari.

A continuació repasarem les diferents característiques:

- El primer que podem veure a l'engegar el terminal és la pantalla d'inici, mitjançant la qual podem accedir a tots els continguts i possibilitats que ens proporciona amb tant sols un clic. Aquesta pantalla és similar a l'escriptori d'un ordinador, les diferents aplicacions estan representades per icones, les quals pots ordenar i crear en nous escriptoris diferents. Aquestes poden ser les aplicacions predeterminades, descarregades des de l'App Store o "Clips Web" (links directes a pàgines web). Amb el botó d'inici es torna directament a l'escriptori.
- La funció de trucada també ha estat millorada per a realitzar-les de forma més fàcil, intuïtiva i ràpida. Es pot consultar també la bústia de veu accedint directament als missatges que vulguem, sense necessitat de escoltar-los tots.
- El correu electrònic funciona igual que en un ordinador, es totalment compatible amb els servidors i proveïdors de correu principals com: MobileMe, Exchange de Microsoft, Yahoo! Mail, Google Mail i AOL i amb pràcticament tots els sistemes de correu estàndard com: IMAP i POP.
- Una de les eines més interessants de les que disposa és el navegador web Safari; potser el més avançat per a navegació web des de mòbil, amb els cercadors Google i Yahoo!. També es capaç de sincronitzar-se amb les "Adreces d'interès" del teu PC o Mac.
- Amb la seva espectacular pantalla de 3'5 polzades i el control "Multi-Touch", l'iPhone és també un fantàstic reproductor de so i vídeo amb les mateixes funcionalitats que el seu germà l'iPod Touch.

- La funció d'escriptura de missatges SMS també ha estat millorada, convertint-la en una de les millors solucions dins del món "smartphones". Gràcies al seu estil tipus xat, al teclat intel·ligent i a la funció d'escriptura predictiva i suggestiva.
- Una de les grans novetats d'aquest dispositiu és el seu avançat sistema de GPS, el qual et permet trobar llocs, obtenir direccions o veure l'estat del trànsit sense la necessitat de cap més aparell. L'iPhone 3G és capaç de donar-te la teva localització via GPS o ve triangulant-la mitjançant Wi-fi o antenes de telefonia mòbil. Tal i com et permet el Google Maps en el teu ordinador el Google Maps desenvolupat per a iPhone et permet canviar entre les diferents visualitzacions com: mapes, imatges per satèl·lit o una funció híbrida d'ambdues.
- El dispositiu te accés directe a l'aplicació iTunes mitjançant Wi-fi o xarxa telefònica, amb la qual et pots descarregar qualsevol dels milions de cançons de que disposa, previ pagament del preu estipulat.
- La App Store és potser la capacitat més espectacular d'aquest dispositiu, permetent la descàrrega de qualsevol aplicació generada per a ell. Amb un sistema similar al del iTunes. Les aplicacions són de tot tipus: jocs, negocis, educació, entreteniment, finances, salut i estat de forma, productivitat, xarxes socials, etc... Totes elles han estat dissenyades per equips de desenvolupadors independents de tot el món per a treure el màxim profit a les característiques del dispositiu com el "Multi Touch", l'acceleròmetre, el wireless i el GPS. Moltes d'elles són gratuïtes i durant el mes febrer de 2009 ja havien superat la xifra de quinze mil.
- La funció de calendari et permet organitzar les teves activitats, fer plans, afegir esdeveniments i tenir-los sincronitzats amb el calendari del teu ordinador.
- També existeix una aplicació relacionada amb el portal d'Internet Youtube, permetent-te crear biblioteques de vídeos preferits.
- Amb la seva càmera de 2 megapíxels i el gestor de fotografies es converteix en el terminal més útil per a la gestió de les mateixes, podent-les sincronitzar de forma automàtica amb el teu ordinador.
- Unes altres aplicacions molt útils són el seguiment de la borsa en temps real, la previsió del temps o un gestor de notes.
- La funció de calculadora et permet treballar tant amb una calculadora senzilla com amb una de científica.

3.2 NECESSITATS DE L'EMPRESA

Per a relacionar el desenvolupament en iPhone amb algun dels projectes en marxa dins del Grup es va triar el Solvay Sustainable. Solvay Sustainable és un projecte engegat per la companyia per a fer-la més

respectuosa amb el medi ambient, amb una sèrie d'objectius molt concrets d'aquí a l'any 2020, els quals veurem més endavant.

3.2.1 SOLVAY SUSTAINABLE



Com resa l'eslògan del projecte "We're all in this together" (Estem tots junts en això), Solvay pren un compromís ferm per al desenvolupament sostenible, el qual és considerat per la companyia com el desafiament més gran amb el que s'haurà d'enfrontar la societat mundial al principi del tercer mil·lenni. El mes de Març del 2007, el comitè executiu va aprovar aquest projecte explícitament integrat dins l'estratègia del Grup.

L'objectiu del Grup Solvay pel desenvolupament sostenible implica la interacció de tots els agents interns en totes les àrees de negoci del grup amb tres premisses bàsiques: creixement econòmic, treball i progrés social i impacte mediambiental. El projecte vol fer progressar la companyia tenint en compte els desafiaments globals (canvi climàtic, subministrament energètic, gestió dels recursos naturals, salut, educació, etc...) i coneixent les creixents expectatives de les entitats involucrades en aquests àmbits (ajuntaments, governs nacionals o regionals, UE, ONG's, etc...), intensificant-hi el diàleg.

3.2.1.1 OBJECTIUS DEL PROJECTE SOLVAY SUSTAINABLE

A continuació repassarem breument els objectius d'aquest projecte ordenats segons el tipus d'entitat relacionada:

- **Inversors:**
 - Utilitzar les eines d'avaluació de sostenibilitat creades per la companyia en el 100% de les innovacions o adquisicions planejades i exportant aquestes eines en el cas de projectes conjunts amb altres societats.
 - Utilitzar les eines d'avaluació de sostenibilitat i els indicadors d'impacte mediambiental creats pel grup per prendre decisions i assignar pressupostos en el 100% de les decisions importants.
 - Tenir projectes d'economització de recursos naturals, que signifiquin més d'un 20% de les propostes d'innovació.
 - Desenvolupar projectes en productes ecològicament eficients, sobretot en el dipòsit de combustible.
- **Societat:**
 - Reduir en un 20% el transport per carretera dels productes a Europa.
 - Reduir la flota de vehicles al voltant d'un 20%.
 - Reduir en un 20% les emissions de CO2 a l'atmosfera, tant de forma directa com indirecta.
 - Reduir un 20% l'energia consumida.
 - Establir auditories de diagnosi de consum d'energia al 80% de les plantes de producció i a tots els centres administratius.

- Aplicar la normativa del Grup sobre l'experimentació amb animals als laboratoris.
- **Clients i proveïdors:**
 - Aplicar auditories al 100% dels proveïdors de serveis logístics.
 - Comunicar als clients la informació del perfil ecològic dels productes.
- **Treballadors i externs:**
 - Establir un diàleg a tots els nivells per a donar suport al projecte.
 - Incloure formació en desenvolupament sostenible a tots els nivells.
 - Afavorir el compromís dels treballadors en el projecte.
 - Introduir noves estructures de protecció pels treballadors enfront dels riscos laborals.
 - Aplicar el sistema d'informació "Health Solvay's Medixis".
 - Aplicar els preparatius per la epidèmia originada per la grip aviar.
- **Comunitats locals:**
 - Reduir el 20% de les emissions contaminants a l'aire.
 - Reduir el 20% de les emissions contaminants a l'aigua.
 - Expandir el sistema d'inspeccions de riscos en les instal·lacions.
 - Obtenir el certificat de qualitat OHSAS 18001 o els seus equivalents en ISO i EMAS.
 - Fer viables les plantes de producció amb risc d'impacte mediambiental.
 - Establir projectes en tots els centres per potenciar la participació dels treballadors en les comunitats locals.
 - Influir en l'opinió d'aquestes societats.

3.2.1.2 MATRIU

Una de les eines creades per a l'avaluació de sostenibilitat que resumeix l'apartat anterior i que proposa el mateix projecte és la matriu 5x5. Aquesta matriu és l'eina encarregada de la representació estratègicopolítica del projecte Solvay Sustainable. Definida l'any 2007 proveeix una estructura a l'estratègia de desenvolupament sostenible del Grup i de totes les seves societats. Aquesta matriu ha de servir de guia en totes les accions econòmiques, socials i de caire mediambiental. En la figura 5 es pot observar la matriu, com es pot veure les files representen els agents implicats en el projecte i les columnes els eixos principals d'aquest projecte. Així doncs en la intersecció d'aquestes files i columnes es tracten els punts bàsics del projecte per a cadascun dels agents involucrats.

		Axis				
		Vision and values Act as a ethical and corporate citizen	Management methods Apply efficient management processes	Products & activities today Supply sustainable, profitable and growing products and services	Future development Lead a world Industrial strategy	Critical risks Manage raising critical Issues
Stakeholders	Investors	Position ourselves as a reference and ethical Industrial Investment	Achieve excellence in managing human, financial and material resources	Prioritize creation of long-term and sustainable value through market leadership	Implement a long-term growth strategy triggered by Innovation and sustainable development	Mitigate risks and related financial impacts
	Society	Practice citizenship beyond our business activities and contribute to scientific knowledge	Reduce global ecological footprint of our production and supply chain activities	Bring products and services presenting environment & health, enhancing global well-being and serving essential needs	Design new products and solutions presenting environment & health, enhancing global well-being, serving essential needs while participating to the evolution of legal framework	Contribute to education, training and employment of young people
	Customers & suppliers	Ensure business ethics	Aim long-term competitiveness and ensure product stewardship	Propose quality, efficient, reliable, documented and validated products and services	Co-develop new sustainable products, services and solutions and their related markets	Take part to the active management of products end-of-life and Anticipate substitution of products
	Personnel & subcontractors	Build a common ethical commitment based on shared values	Guarantee fair labour conditions, safe working conditions, empowerment and career management	Expand multidisciplinary, diversity, network / partnership practices and competences in sustainable development	Favour creativity and innovation	Minimize critical risks and related human impacts : accidents, occupational diseases, layoffs, loss of expertise
	Local communities	Raise community dialogue and participation to local life (neighbours, authorities, associations, ...)	Optimize environmental performances	Contribute to local wealth : employment, salaries, purchase of local goods and services	Strengthen local economical development : clusters of economic activities, infrastructures and equipments, competences	Protect neighbours and their living environment : health, environment, employment, major risks prevention

Figura nº 5.

3.3 EL PROJECTE

La realització del projecte en iPhone vol col·laborar en la divulgació del projecte Solvay Sustainable; fent-lo més accessible als diferents àmbits de l'empresa, millorant la comunicació dels departaments o plantes de producció entre elles i amb els seus treballadors; afavorint així l'acompliment dels objectius marcats de forma més ràpida i satisfactòria.

3.3.1 LA IDEA

La idea concreta del projecte és la de crear una aplicació per a iPhone on es pugui representar la matriu 5x5 per a cadascun dels diferents centres dels que disposa Solvay, i que a través d'aquesta representació s'accedeixi a contingut generat per a les diferents cel·les o temes a tractar. Aquesta informació ha de ser correctament visualitzable i útil per a les persones que en faran ús. El programari generat també ha de ser actualitzable i reutilitzable.

L'aplicació s'ha d'integrar dins d'un sistema complet d'actualització i distribució de la mateixa en el si de l'empresa, mitjançant el qual les persones que en facin ús disposin d'un entorn senzill, dinàmic, segur i còmode de treball que els permeti modificar els continguts mostrats i serveixi de base per a futurs desenvolupaments; tant en el marc del projecte Solvay Sustainable com en aplicacions i serveis encaminats a la integració de la tecnologia iPhone en el si de la companyia.

Aquest projecte també vol aprofundir en el coneixement de la tecnologia desenvolupada per Apple en el cas concret d'iPhone i conèixer les seves possibles aplicacions. Així mateix poder aplicar aquest coneixement en trobar solucions cada cop més eficients en el desenvolupament global de la companyia.

3.4 RESUM

En aquest capítol s'ha aprofundit en els diferents conceptes que formen part del projecte. Com poden ser la situació prèvia a la realització del mateix i com s'ha arribat a aquesta situació; el terminal amb la seva tecnologia associada i la revolució que ha suposat dins del propi mercat de telefonia mòbil; com aquesta revolució pot afectar la vida de les persones i convertir el terminal en la ja anomenada "quarta pantalla" (després del cinema, la televisió i l'ordinador). També s'ha aprofundit en el projecte Solvay Sustainable, amb el qual s'ha conjugat el projecte iPhone amb el projecte Solvay Sustainable i tot això per a poder enllaçar amb els següents capítols.

4. ESTUDI DE VIABILITAT

4.1 INTRODUCCIÓ

El projecte iSustainable és la realització d'una aplicació per al dispositiu mòbil iPhone a l'empresa Solvay, mitjançant la qual es pugui visualitzar informació de les diferents plantes de producció utilitzant una representació estàndard en forma de matriu. També s'ha generat un sistema d'actualització de la base de dades directament descarregable des del dispositiu i emplaçada en un servidor web, el qual permet mantenir la informació al dia, a través d'una aplicació web operativa des de qualsevol terminal amb connexió a la xarxa. Per aquest projecte s'ha utilitzat una metodologia de prototip, ja que es tracta d'un sistema totalment nou i desconegut per a la companyia; el qual s'ha hagut de desenvolupar completament des de zero.

En aquest estudi de viabilitat analitzarem la viabilitat tècnica, econòmica, operativa i legal i les diferents alternatives al projecte. També avaluarem les garanties d'acabar el projecte amb èxit, els beneficis que aquest aporta a l'empresa i proporcionarem els arguments per decidir quina solució pot ser la millor resposta per aquest projecte.

4.1.1 AVANTATGES

- Conèixer la tecnologia iPhone.
- Aprendre a desenvolupar aplicacions per entorns Apple en general i iPhone en particular.
- Integrar una nova tecnologia a la empresa utilitzant-la per l'acompliment d'uns objectius de sostenibilitat amb el medi ambient.
- Proporcionar la informació a la gent adequada d'una forma còmode, podent-la consultar en qualsevol moment.

4.1.2 INCONVENIENTS

- Necessitat d'actualització de la base de dades.
- Incorporació de terminals iPhone.
- Integració de tecnologia Apple en la companyia.
- Formació de personal per a l'actualització.

4.2 OBJECTE

4.2.1 SITUACIÓ A TRACTAR

Aquest projecte pren el seu significat per la voluntat constant d'innovació del departament d'informàtica (SIS) de l'empresa Solvay i concretament de la seva divisió a Espanya. Aquest departament impulsa diferents projectes informàtics i de comunicacions d'implantació arreu del món en els quals s'intenta donar un pas endavant i estar atents a les noves tecnologies que van apareixen per en el cas de ser interessants implantar-les a l'empresa.

4.2.1.1 MOTIUS I NECESSITATS

- Innovar.
- Investigar el desenvolupament d'aquesta nova tecnologia i veure les seves possibles aplicacions.
- Dotar els directius d'un sistema còmode, fàcil i ràpid de consulta de dades per a la seva feina.
- Treure el màxim profit a una tecnologia en expansió.
- Convertir l'iPhone en una eina de treball més completa.
- Donar valor afegit a la plataforma.

4.2.2 PERFIL DE L'USUARI

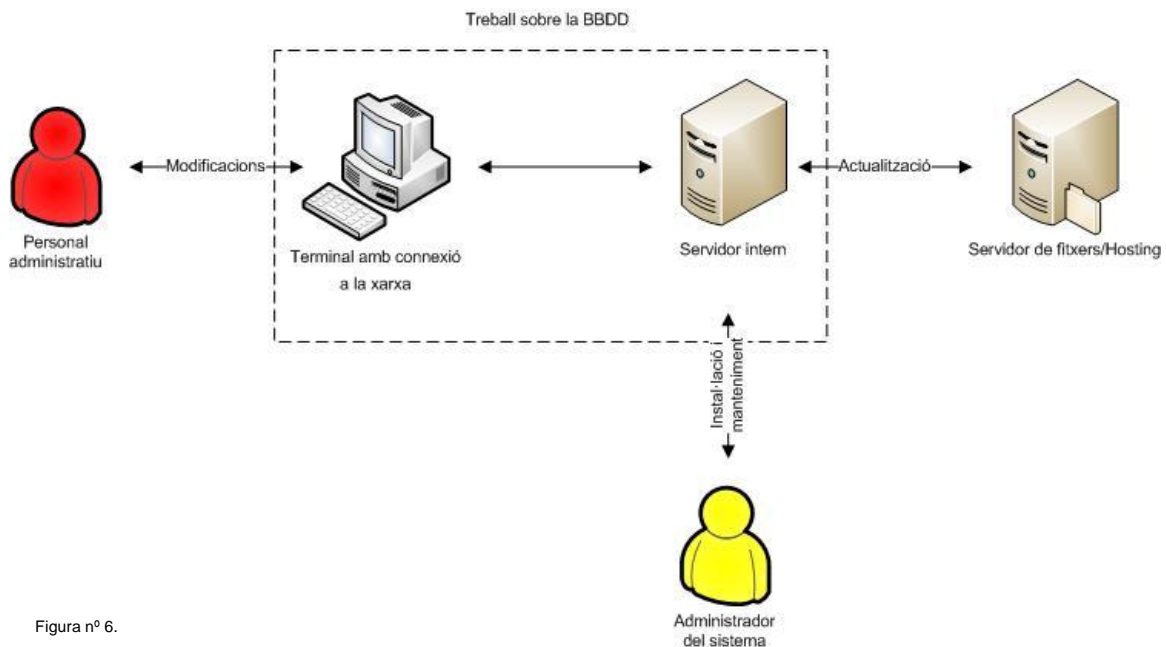
En aquest sentit existeixen tres perfils clarament diferenciats: per una part tenim l'usuari de l'aplicació iPhone, per l'altra part l'usuari del programari d'actualització i una tercera seria la persona o persones encarregades de l'administració d'ambdues aplicacions. Tot i que en un principi aquests usuaris seran persones diferents en algun cas es poden superposar en la mateixa persona total o parcialment.

4.2.2.1 USUARI DEL PROGRAMARI D'ACTUALITZACIÓ

Aquesta persona serà l'encarregada d'actualitzar la base de dades, que posteriorment es mostrarà a l'aplicació. Haurà de tenir un nivell d'usuari de coneixements informàtics, amb el qual no ha de tenir cap problema per a importar, publicar o modificar la BBDD. Normalment els encarregats de realitzar aquesta tasca seran personal administratiu o secretariat.

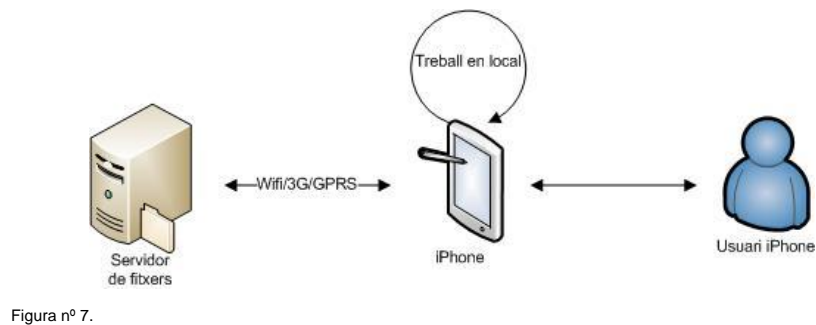
4.2.2.2 ADMINISTRADOR

La persona encarregada d'aquesta tasca haurà de tenir uns coneixements informàtics avançats o de nivell programador. Haurà de conèixer mínimament el projecte, l'aplicació, la distribució d'aplicacions iPhone, bases de dades, SQL, programació web i gestió de servidors web. Ja que serà la qual haurà de resoldre possibles problemes de funcionament, orientar en cas de dubte els diferents usuaris i en definitiva vetllar pel correcte funcionament de tot el sistema.



4.2.2.3 USUARI DE L'APLICACIÓ IPHONE

Aquesta persona serà la propietària del terminal mòbil i la que traurà profit de l'aplicació. Haurà de tenir un coneixement suficient del funcionament del terminal i de les seves aplicacions, per la qual cosa s'haurà previst la formació adient. Ja que aquesta funcionarà de forma similar a les ja existents en el App Store. Les persones que disposaran d'aquest dispositiu i de l'aplicació en concret seran tan directius com caps de departament o directors de planta.



4.2.3 OBJECTIUS

- Integrar la tecnologia iPhone a l'empresa.
- Millorar la comunicació entre els diferents departaments.
- Desenvolupar un sistema per a la representació de la matriu, mitjançant el qual s'hi puguin introduir continguts i es puguin consultar.
- Aconseguir una interfície fàcilment manejable i intuïtiva.
- Realitzar una BBDD ràpidament consultable i robusta.
- Fer possible l'actualització on-line.

- Crear tot un entorn per a poder editar de forma senzilla i segura la BBDD, on qualsevol usuari sigui capaç de interactuar sense problemes.

4.3 DESCRIPCIÓ DEL SISTEMA

4.3.1 DESCRIPCIÓ

- Integrar la tecnologia iPhone a l'empresa.
 - o Per aconseguir aquest objectiu els directius han de poder comprovar que la utilització d'aquest dispositiu els hi facilita la seva tasca, per tant han de aconseguir-se en gran part la resta d'objectius que analitzarem a continuació.
- Millorar la comunicació entre els diferents departaments.
 - o Aconseguir que les dades importants arribin realment a les persones, que en puguin disposar en qualsevol moment i que la seva consulta no sigui una tasca feixuga. En aquest objectiu pot ser de gran utilitat les interfícies per treballar amb la pantalla tàctil.
- Desenvolupar un sistema per a la representació de la matriu que s'ha introduït en el capítol anterior, mitjançant el qual s'hi puguin introduir continguts i es puguin consultar.
 - o Introduir una imatge de la matriu 5x5 i aconseguir-ne una correcta visualització.
 - o Mapejar la matriu per obtenir el punt exacte on a pitjat l'usuari per a poder accedir a una o altra informació.
 - o Realitzar la consulta a la BBDD i mostrar les dades per pantalla.
- Aconseguir que sigui la interfície fàcilment manejable i intuïtiva.
 - o Realitzar un disseny molt intuïtiu on l'usuari s'hi trobi còmode.
- Realitzar una BBDD ràpidament consultable i robusta.
 - o Fer un estudi molt acurat de l'aplicació que es vol realitzar per aconseguir una BBDD on s'obtinguin respostes ràpides a les consultes.
- Fer possible l'actualització on-line.
 - o Implementar una solució mòbil per a la descàrrega de les actualitzacions de la BBDD, on mitjançant una connexió wifi, 3G o GPRS que et permet el terminal, aquest es connecti a un servidor i es descarregui la nova BBDD.
- Crear un entorn per a poder editar de forma senzilla i segura la BBDD, on el personal de nivell usuari sigui capaç de interactuar sense problemes.
 - o Desenvolupar un entorn senzill on els usuaris no tinguin dificultats amb el funcionament, aquesta part del programari ha de permetre que persones sense cap coneixement de programació o de BBDD siguin capaços d'actualitzar la BBDD.

4.3.2 RECURSOS

4.3.2.1 HARDWARE

Desenvolupament:

- Portàtil Apple amb sistema operatiu Mac OS X Leopard 10.5.
- iPhone o iPod Touch.

Producció:

- Ordinador amb processador Pentium 4 CPU GHz i 512Mb de memòria RAM.
- Hosting web virtual Linux Sync 600L.
- iPhone.
- Servidor HP Proliant DL 380 G5

4.3.2.2 SOFTWARE

Desenvolupament:

- Sistema operatiu Mac OS X 10.5 o superior.
- SDK per iPhone OS 2.1.
- Llicència de desenvolupament en iPhone "Standard Program".
- Mozilla Firefox 3.0.
- SQLite Manager Mozilla Firefox Add-on 0.4.
- Python 2.5.
- Turbogears 1.0.
- Notepad++ 5.1.
- SQLAlchemy 0.5.

Producció:

- Sistema Operatiu Microsoft Windows XP Profesional versió 2002.
- Màquina virtual servidor VMWare ESX.
- Sistema Operatiu Ubuntu per a servidor.

5.3.2.3 RECURSOS HUMANS

En l'apartat de recursos humans és necessari la participació de dos perfils diferents:

- Analista/Cap de projecte a temps parcial.
 - o Responsable de supervisar el correcte desenvolupament del projecte segons les premisses marcades.
- Programador/Becari.
 - o Encarregat de la realització del projecte.

4.3.3 AVALUACIÓ DE RISCOS

En aquest projecte existeixen dos riscos potencials, són els següents:

- Cancel·lació del projecte per part de l'empresa, degut al moment econòmic complicat que travessa l'economia mundial i espanyola.
- Cancel·lació del projecte per part de l'empresa, degut a que el projecte i/o la tecnologia no compleix els requeriments suficients per a ésser viable.

4.3.4 ORGANITZACIÓ DEL PROJECTE

4.3.4.1 ETAPES I METODOLOGIA DE DESENVOLUPAMENT

El continguts d'aquest apartat coincideixen amb els que s'han introduït en l'apartat "Organització del projecte" del capítol Introducció.

4.3.4.2 ANÀLISI DE COSTOS I BENEFICIS

Cost de material:

Recurs	Cost total
Apple Mac Book	1.149€
iPhone	700€
Amortització connexió ADSL	*
Amortització Servidor	*
Mac OS X 10.5	0€
SDK per iPhone OS 2.1	0€
iPhone Standard Program	70€
Mozilla Firefox 3.0	0€
SQLite Manager Firefox 0.4	0€
Python 2.5	0€
Turbogears 1.0	0€
Notepad++ 5.1	0€
SQLAlchemy 0.5	0€
Amortització PC	*
Amortització MSOffice	*
Total	1.919€

Figura nº 8.

En les diferents files on hi trobem un asterisc (*) no són computables, ja que s'ha utilitzat l'estructura de la companyia per a la seva realització.

Cost personal:

Tasca	Treball Analista	Treball Becari	Cost
Aprenentatge iPhone		60h	340,8€
Entrevistes	5h	5h	403,4€
Estudi dels requeriments	5h	5h	403,4€
Disseny aplicació iPhone i BBDD	2h	10h	206,8€
Implementació		80h	454,4€
Fase de proves		20h	113,6€
Estudi d'alternatives	5h	5h	403,4€
Aprenentatge Turbogears i Python		60h	340,8€
Disseny del programari	2h	10h	206,8€
Implementació		90h	511,2€
Fase de proves		40h	227,2€
Documentació i manuals d'ús		70h	397,6€
Total	19h	455h	4.009,4€

Figura nº 9.

Recurs	Cost(h)
Analista	75€
Becari	5,68€

Beneficis:

Els beneficis no es poden compatibilitzar quantitativament en aquest moment. Ja que un dels objectius principals d'aquest projecte és la voluntat d'inversió i desenvolupament constant en noves tecnologies per part de la SIS. El qual vol donar-li valor afegit a una plataforma mòbil en ple creixement, de màxima actualitat en aquests moments, que els directius demanen cada cop més. Els beneficis no es comptabilitzen quantitativament amb ROI enfront inversió, sinó també en el valor afegit que aporta a la capa directiva de la companyia. Principal objectiu del projecte.

4.4 PLANIFICACIÓ DEL PROJECTE

Per a la planificació del projecte s'ha utilitzat un model lineal, amb una estructura dividida en mòduls. El software s'ha dissenyat en dues etapes diferents, una per a cada aplicació. La figura 10 generada amb MSPProject s'hi poden observar les diferents tasques i com han avançat en el temps:

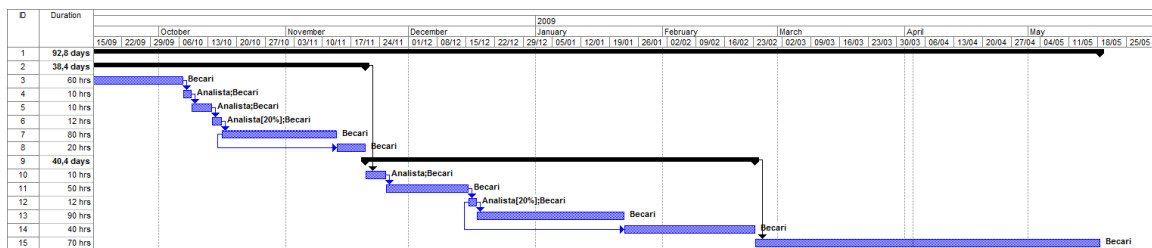


Figura nº 10.

4.5 CONCLUSIONS

L'elaboració d'aquest projecte introdueix a l'empresa en una nova tecnologia, adquirint nous coneixements. Permet als seus directius disposar d'una eina de treball que els hi facilita les comunicacions i fa els seus viatges i reunions més còmodes; ja que no han d'emportar-se còpies en paper durant els vols i poden consultar la informació en qualsevol moment. Estalvia paper a la companyia contribuint a millorar el medi ambient i treball a les persones encarregades de preparar les reunions.

Les aplicacions creades han estat realitzades pensant d'ésser fàcilment mantenibles i de baix cost. Tant el cost de realització com de manteniment d'aquestes aplicacions és considerablement baix, degut a la utilització de programari lliure i l'aprofitament d'estructures ja existents en la companyia. També s'han de considerar els beneficis a llarg termini que aquests tipus de projectes tenen en la imatge que vol oferir la companyia a la societat, invertint la investigació de joves universitaris. Per tots aquests motius el projecte és considerat viable.

5. ANÀLISI I DISSENY

5.1 DISSENY TRONCAL

Per a dissenyar el projecte s'ha utilitzat una metodologia de prototip. En la qual s'ha anat evolucionant a partir d'una idea inicial, generant un model primari i desenvolupant cadascun dels mòduls per separat fins a tenir-ne una versió de proves amb la que posar en comú la resta del conjunt. Com podem veure en la figura 11.

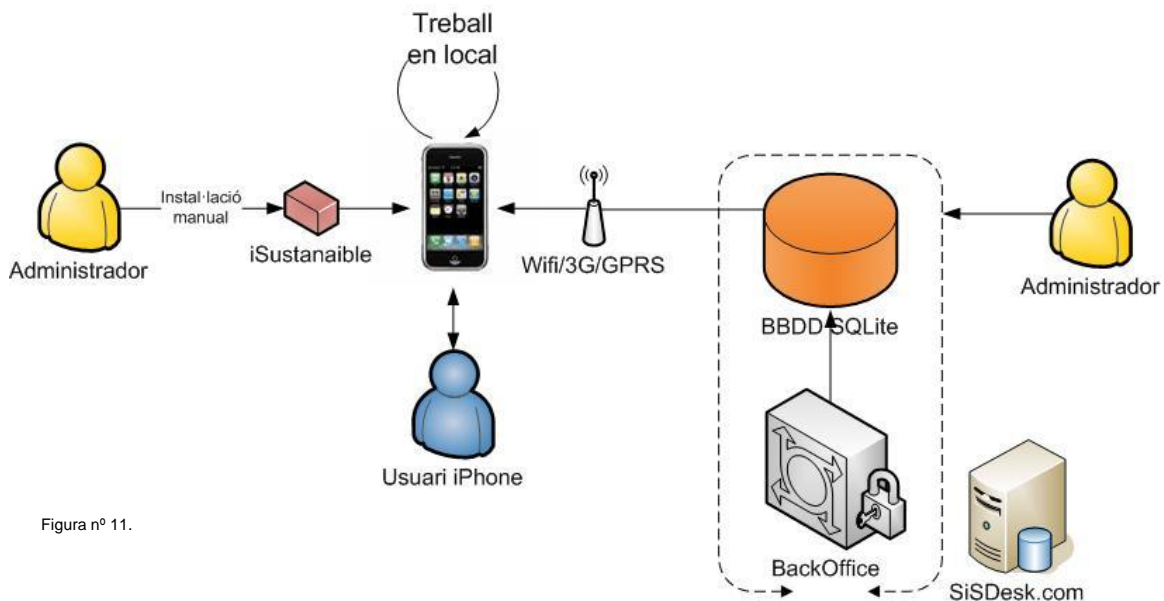


Figura nº 11.

Aquest disseny primari parteix de la idea de generar un sistema per actualitzar l'aplicació, creant un "back-office" o aplicatiu de suport, mitjançant el qual es podran modificar els continguts mostrats en el programari i obtenir-los directament al dispositiu. Un dels aspectes que ha marcat més el disseny del sistema ha estat la impossibilitat de estar sempre connectats a la xarxa, això ha provocat tota una sèrie de decisions de disseny encaminades a dissenyar una aplicació "off-line" (que funcioni sense una connexió a Internet en tot moment).

Aquesta decisió es va prendre després de sopesar els avantatges i inconvenients que genera el crear una aplicació en local o en remot. Els principals avantatges de generar l'aplicació en remot són:

- Minimitzar el temps d'aprenentatge.
- Augmentar les possibilitats de l'aplicació.
- Poder consultar les dades en temps real.

El temps d'aprenentatge es minimitzaria ja que no seria necessari aprendre en profunditat el funcionament del llenguatge Objective-C i de les eines de desenvolupament d'Apple; degut a la possibilitat de crear una aplicació web en un format adaptat al navegador integrat d'iPhone i poder-la

visualitzar gràcies a aquest navegador. Aquesta opció també incrementava considerablement les possibilitats, ja que això permetia treballar de forma més independent, escollint les eines i els llenguatges més adients en cada cas; sense les restriccions prestacionals imposades per un terminal mòbil. També ens permetia la possibilitat de actualitzar les dades en temps real.

Però tots aquests avantatges han estat desestimats, degut a un requeriment. Aquest requeriment és el de poder consultar l'aplicació en tot moment. Malgrat la seva capacitat de connexió Wi-fi i 3G, la impossibilitat d'obtenir una connexió gratuïta d'aquest tipus en qualsevol lloc o país on es pugui trobar l'usuari van influir decisivament en el disseny del projecte i el van empènyer cap a un aplicatiu local.

A continuació introduïrem el diagrama de casos d'ús del projecte en la figura 12:

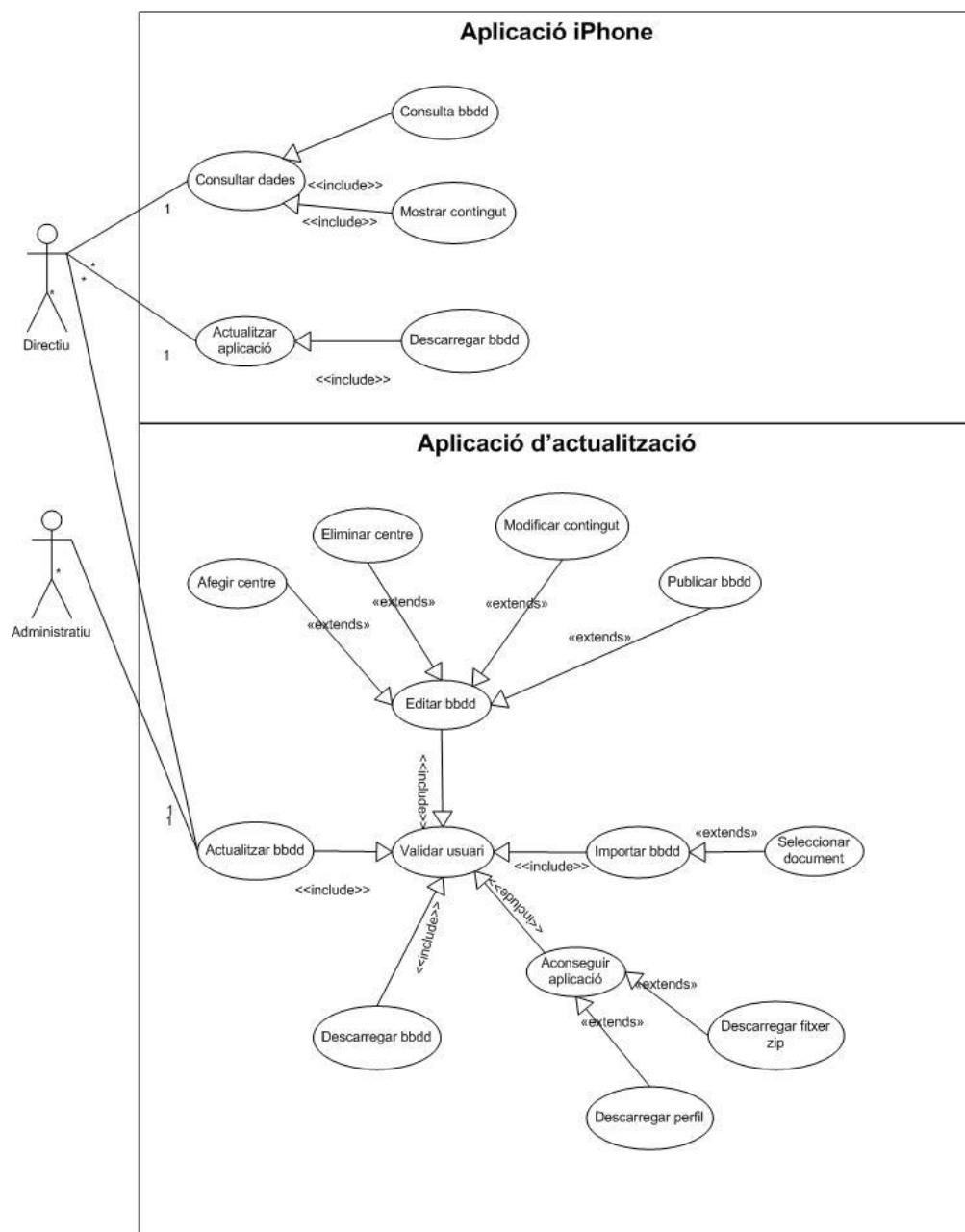


Figura nº 12.

En el diagrama de casos d'ús de la figura anterior es poden observar els diferents perfils d'usuari que en faran ús. Identifiquem dos tipus diferents d'usuari, el directiu i l'administratiu. El directiu és l'usuari que utilitzarà l'aplicació iPhone i per al que està dissenyat tot el projecte, però també podrà tenir accés a l'aplicació per a actualitzar la BBDD. L'administratiu representa algun dels usuaris encarregats d'actualitzar la BBDD, però no emprarà l'aplicació per l'iPhone al no disposar d'aquest terminal. També es poden identificar dos sistemes completament diferenciats, l'aplicació estrictament dissenyada per a l'iPhone i la dissenyada per actualitzar la BBDD. Com es pot veure en la figura, l'aplicació per iPhone consistirà en dos accions principals, consultar dades i actualitzar aplicació. En el cas de consultar dades l'usuari accedirà als continguts de la BBDD i els podrà visualitzar. Per la part d'actualitzar l'aplicació l'usuari podrà descarregar-se una nova BBDD.

Pel que fa al sistema d'actualització de la BBDD, existeixen quatre accions diferenciades: editar centre, importar BBDD, exportar BBDD i obtenir l'aplicació (inclou descarregar fitxer zip i descarregar perfil). En l'acció validar usuari, aquest haurà d'identificar-se mitjançant un nom usuari i una clau. A editar BBDD s'hi podran afegir centres, eliminar centres o modificar contingut (crear, eliminar o modificar el contingut generat per a cadascun dels centres) i a publicar BBDD es publicarà la BBDD al servidor per a poder descarregar la nova versió al dispositiu mòbil. Importar BBDD permetrà penjar fitxers directament el directori secundari del servidor, tot i que aquesta funcionalitat no serà plenament operativa en aquesta primera versió de l'aplicatiu. Exportar BBDD, en aquesta funcionalitat es podrà obtenir directament el fitxer de la BBDD. I per últim descarregar fitxer zip i perfil, amb la qual cosa podràs obtenir l'aplicació iPhone en qualsevol ordinador amb la que després poder instal·lar-la en el terminal.

Una vegada comentat el diagrama de casos d'ús entrarem en més detall descrivint el disseny de cada part del sistema. A continuació anem a centrar-nos en el disseny de la aplicació per a l'iPhone, aquest disseny i les seves possibilitats ens marcaran el camí a seguir en la resta del projecte (BBDD i aplicació d'actualització).

5.2 DISSENY DE LA APLICACIÓ IPHONE

L'aplicació dissenyada per a l'iPhone té en compte una sèrie de conceptes: aconseguir un programari senzill i molt intuïtiu, executar l'aplicació completament en local (en el mateix dispositiu), crear un entorn adequat a la representació de la matriu i a la resta de continguts generats, actualitzar la informació mostrada per l'aplicació a distància i sense la necessitat d'instal·lar-la de nou en el dispositiu, visualitzar les dades de diferents centres amb el mateix programari.

Es parteix d'una idea inicial de tres capes, en la qual hi apareixeran els següents continguts:

- 1- Una llista amb els diferents centres.
- 2- La matriu 5x5 del projecte Solvay Sustainable.
- 3- Informació relacionada amb la cel·la o fila de la matriu i el centre seleccionat.

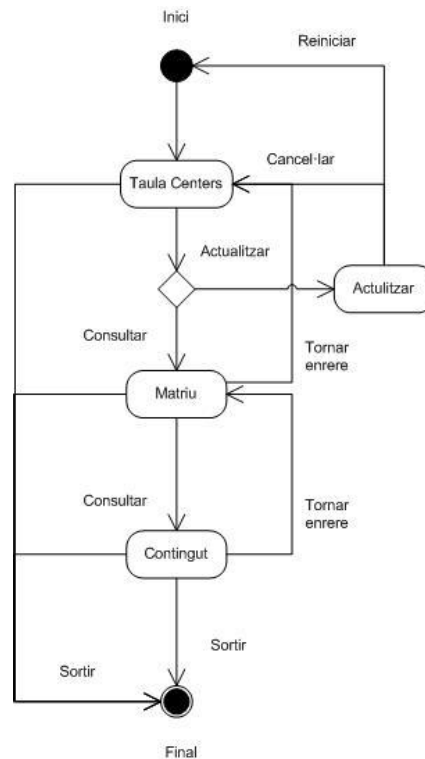


Figura nº 14.

5.3 DISSENY DE LA BBDD

El disseny de la BBDD és potser la part més important de tot el disseny. La BBDD es dissenya tenint en compte el tipus d'aplicació que es vol realitzar i les funcions d'aquesta. La importància de la BBDD radica en que és un element essencial de qualsevol programari de tipus dinàmic; la qual serveix per a consultar dades emmagatzemades per a mostrar-les per pantalla i/o realitzar operacions amb elles i guardar noves dades introduïdes per l'usuari per a la seva posterior utilització. La posterior eficiència de la aplicació, la facilitat per a construir el llenguatge i la seva ràpida consulta en depenen. És per això que resulta de vital importància dedicar una càrrega important de temps en el disseny de la mateixa per a obtenir-ne un el més òptim possible.

La BBDD dissenyada pel projecte està formada per quatre taules: "Center", "Content", "Image" i "ImageArea". Aquestes quatre taules són les indispensables per el correcte funcionament del programari, el disseny ha estat pensat per a ésser el més simple possible i d'aquesta manera estalviar-nos taules i poder realitzar consultes mes senzilles i fàcils d'implementar, sense que això afecti al rendiment de la aplicació. També s'ha intentat que els noms fossin el més descriptius possible.

La taula Center és on s'emmagatzemen els diferents centres de producció que després seran visualitzats en la primera pantalla del programari. La taula esta formada per tres camps: "IDCenter", "Name" i "Comment". El camp IDCenter és de tipus enter, és l'identificador dels diferents centres i a la vegada fa la funció de clau primària de la taula, també està a la taula Content i la seva relació és 1-N (1 centre - N continguts). El camp Name és de tipus text i és on s'emmagatzemen els diferents registres amb el nom de

cada centre. El camp Comment també és de tipus text i s'utilitza per a guardar els registres referents al país on es troba el centre.

La taula Content és la principal de la BBDD i el seu objectiu és fer de nexa d'unió entre les imatges, el mapeig de la matriu i el centre. Està formada pels camps "IDContent", "Comment", "IDCenter", "HTMLContent", "IDImage" i "IDImageArea". El camp IDContent és de tipus enter, realitza la funció d'identificador dels continguts i a més a més és clau primària de la taula. La seva relació és de 1-1..N (1 contingut - entre 1 i N cel·les de la matriu). El camp Comment és de tipus text i s'utilitza per a introduir-hi l'agent implicat en la fila de la matriu. El camp HTMLContent és de tipus caràcter i és un camp addicional per a possibles ampliacions futures del programari.

La taula Image és on s'emmagatzemen les diferents imatges que formen part del contingut de cada centre. La taula està formada pels camps "IDImage", "Image" i "Comment". El camp IDImage és de tipus enter, és l'identificador de les diferents imatges i fa la funció de clau primària de la taula. Està present en la taula Content i la seva relació és 1-1 (1 imatge – 1 contingut). El camp Image és de tipus blob (que és un tipus per emmagatzemar fitxers) i és on es guarden les imatges. El camp Comment és de tipus text i també és addicional per a possibles ampliacions.

La taula ImageArea és implementada pel mapeig de la matriu. Està formada pels camps "IDImageArea", "X1", "X2", "Y1", "Y2", "Comment" i "ImagePosition". El camp IDImageArea és de tipus enter, és l'identificador de les diferents cel·les de la matriu i realitza la funció de clau primària. Està present també a la taula Content i la seva relació és 1-1 (1 contingut – 1 cel·la). El camps X1, X2, Y1 i Y2 són les coordenades x i y dels dos punts necessaris per a definir l'àrea d'un rectangle, són de tipus decimal. El camp Comment és de tipus text i s'ha introduït per a possibles ampliacions del programari. El camp ImagePosition és de tipus enter i representa la posició del dispositiu respecte la verticalitat.

La figura 15 mostra el disseny de la BBDD:

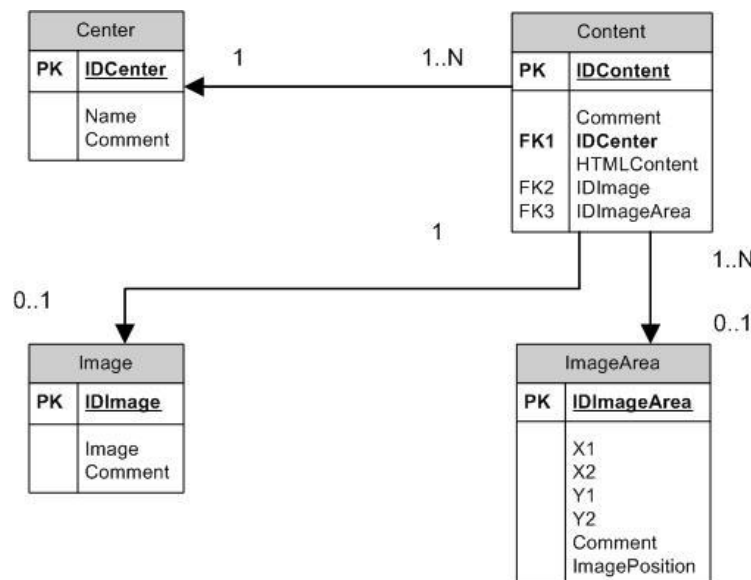


Figura nº 15.

5.4 DISSENY DE LA APLICACIÓ D'ACTUALITZACIÓ

La aplicació d'actualització s'ha dissenyat per què qualsevol persona externa al projecte i sense coneixements de programació en Objective-C o SQL, sigui capaç de realitzar modificacions sobre la BBDD. D'aquesta manera les dades poden ser modificables per un nombre més gran de persones i es pot alliberar d'aquesta feina el departament d'informàtica, facilitant-ne la tasca. Per a realitzar aquesta aplicació s'ha pensat en un entorn web al que s'accedirà mitjançant una connexió a la xarxa a través d'un navegador. Aquesta decisió es fonamenta principalment en dues raons, la primera i més important és que està pensat com un projecte global, al qual s'hi pugui accedir des de qualsevol lloc de la empresa només amb una connexió a la xarxa; la segona evitar instal·lar programari addicional així com aprofitar les sinergies amb el software lliure.

Per a poder accedir a la aplicació s'haurà de conèixer un nom d'usuari i una paraula clau, el que permetrà restringir l'accés a la aplicació a les persones no autoritzades. Aquestes persones hauran de posar-se en contacte amb l'administrador de l'aplicació per a poder aconseguir l'usuari i la paraula clau. Un cop introduïdes aquestes dades l'usuari accedirà a una pantalla principal o menú la qual vertebrarà tot el programari. En ella es podrà visualitzar els diferents mòduls.

L'aplicació és dividirà en tres mòduls diferents, segons l'acció que es vulgui realitzar sobre la BBDD. Els mòduls són: editar, importar la BBDD o aconseguir la aplicació:

- Editar BBDD: en aquest mòdul es visualitzaran els diferents centres que existeixen en un moment determinat en la BBDD. Podran afegir-se o eliminar centres, es podrà seleccionar un centre concret i editar-lo o publicar els canvis realitzats (realitzar una còpia de la BBDD modificada en l'emplaçament disposat per a la seva posterior descàrrega en el terminal mòbil).
 - En l'opció editar centre del mòdul s'accedirà a una nova pantalla on podrem visualitzar la matriu 5x5 del projecte Solvay Sustainable, la qual es mapejarà per a poder seleccionar individualment cadascuna de les cel·les de la matriu (tal i com es realitza en la aplicació iPhone) i editar, modificar o suprimir el seu contingut.
 - Un cop seleccionada la cel·la concreta es visualitzarà la imatge introduïda en aquella secció o un espai en blanc en cas contrari. Fet que permetrà eliminar el contingut o afegir-ne de nou.
- Importar BBDD: En aquest mòdul l'usuari podrà descarregar-se la BBDD que en aquell moment estigui penjada en el servidor per a poder tenir-ne una còpia. Així l'usuari podrà tenir una còpia de seguretat en cas de modificació.
- Aprovisionament de l'aplicació: En aquest mòdul l'usuari podrà descarregar-se dos fitxers diferents, un fitxer de tipus *.zip (amb la aplicació iPhone) i una altre de tipus *.mobileprovision (amb el perfil d'usuari necessari ha instal·lar en l'iPhone pel funcionament de la aplicació).

A la figura 16 podem observar el diagrama d'estats de la aplicació:

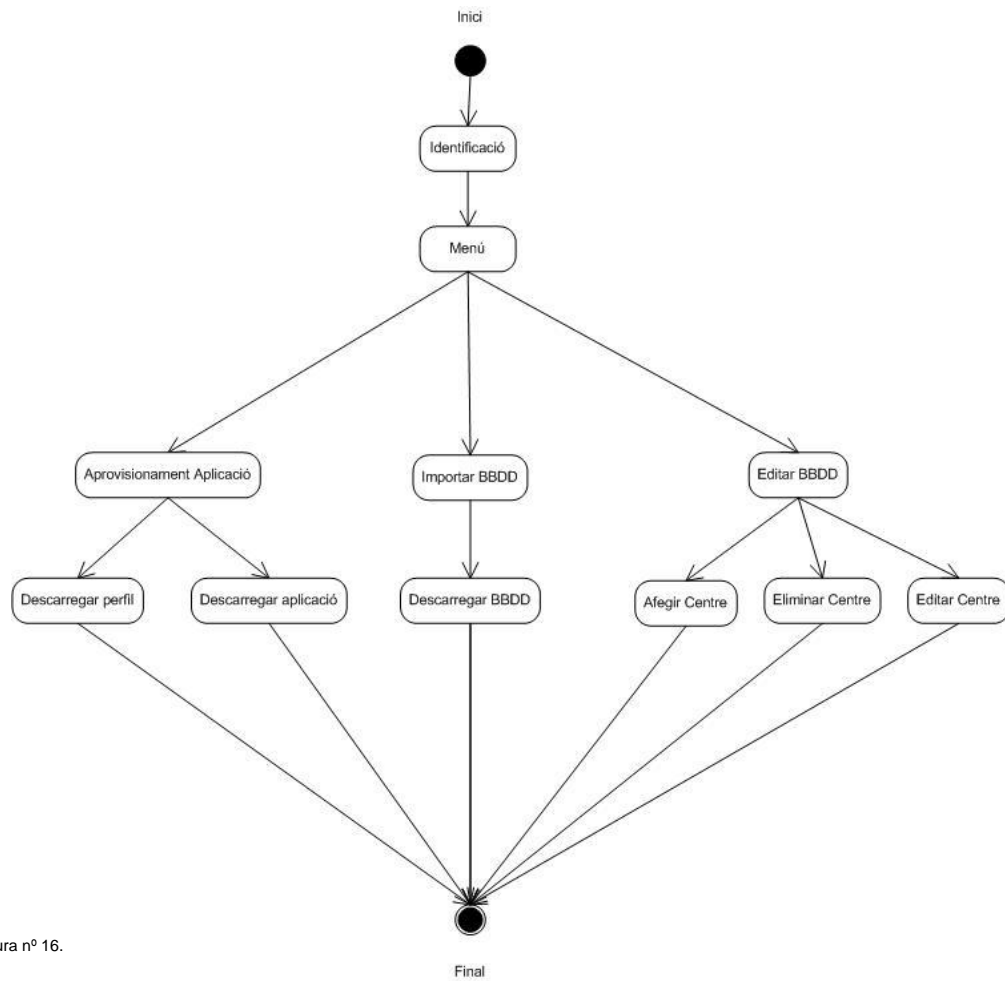


Figura nº 16.

5.5 RESUM

En aquest capítol s'ha tractat el disseny troncal del projecte, partint d'una idea inicial, i s'ha anat evolucionant en el disseny de les diferents parts que el componen. També s'han tractat les diferents decisions de disseny, amb els seus motius i conseqüències. El primer apartat introdueix la idea inicial i la visualitza mitjançant un gràfic d'estructura. Posteriorment un gràfic de casos d'ús del sistema mostra de forma visual la interacció dels diferents perfils d'usuari amb el sistema. El segon i el quart apartat tracten el disseny de les aplicacions per l'iPhone i per d'actualització respectivament, descrivint com es vol que siguin i visualitzant els diferents estats pels que passaran les mateixes. El tercer tracta el disseny de la BBDD, essent aquest molt important en el totalitat del conjunt i visualitzant-t'ho mitjançant el gràfic adjunt amb les diferents taules, relacions entre elles i els seus identificadors.

6. ENTORN DE DESENVOLUPAMENT

L'elecció de la plataforma iPhone suposava un repte afegit en el desenvolupament del projecte, ja que el seu coneixement era pràcticament inexistent abans de la realització del mateix. Durant la fase d'aprenentatge la pròpia plataforma estava en fase "beta" per als propis desenvolupadors. S'ha anat creixent amb la pròpia plataforma i han hagut de resoldre molts problemes per a desenvolupar en un entorn inestable i molt canviant d'una versió beta a la següent.

Aquestes mancances s'han anat minimitzant fins al punt de tenir uns coneixements suficients per a poder crear una aplicació que cobris les necessitats de l'empresa. A posteriori també ha calgut anar triant els llenguatges i les eines de desenvolupament que ens han semblat més adequades per a la realització del back-end (o part del servidor, emprades en l'actualització de la BBDD del programari).

El propi llenguatge de programació ja era un desafiament en si mateix, degut a les diferències existents amb d'altres llenguatges més comuns o amb els impartits a la universitat, tot i que mantenint unes similituds amb llenguatges com el C o el Java. Aquest llenguatge, anomenat Objective-C és propietari d'Apple i és l'utilitza't per a realitzar els programes de la família Apple. Dins d'aquest entorn existeix un programari de desenvolupament pensat especialment per a desenvolupar tot tipus d'aplicacions. A continuació desenvoluparem els elements principals de programari emprats en el projecte, amb la seva història, característiques principals i les raons per les quals els em triat en el cas de ser necessari.

6.1 OBJECTIVE-C

L'Objective-C³⁴ és un llenguatge orientat a objecte (OO) creat com un superconjunt de C, però que implementa un model d'objectes semblant al de Smalltalk. En l'actualitat s'utilitza bàsicament en MAC OS X, iPhone OS i GNUstep. Originalment va ser escrit per Brad Cox i Tom Love a la seva corporació StepStone a la dècada dels '80.

6.1.1 HISTÒRIA

A principis dels 80's, el software es desenvolupava mitjançant programació estructurada; però aquesta comportava, ja en aquells moments, nombrosos problemes; degut a basar-se en subdividir els programes en conjunts més petits per a facilitar el seu desenvolupament. El increment constant de la complexitat dels problemes a resoldre suposava un nou desafiament per a la informàtica, la qual cosa produïa que la programació estructurada fos cada cop menys útil per resoldre aquests inconvenients, creant una gran repetició en el codi i impossibilitant la seva reutilització.

³ (Apple Team, 2008)

⁴ (Anònim, Wikipedia.org, 2008)

Cox i Love que ja coneixien el programari Smalltalk amb anterioritat van començar a desenvolupar un nou llenguatge de programació modificant el compilador de C i afegint-hi capacitats ja introduïdes per Smalltalk. Aviat van tenir una extensió per afegir la programació OO al C, la qual van anomenar “OOPC” (Object-Oriented Programming in C). Posteriorment van formar una empresa anomenada PPI (Productivity Products International) per vendre el seu producte, un compilador de C amb una sèrie de llibreries potents per la època.

El 1988, NeXT, la companyia que va fundar Steve Jobs (actual president i un dels fundadors d’Apple), després de deixar Apple, va llicenciar un Objective-C de Stepstone (el propietari de la marca Objective-C) per treure al mercat el seu propi compilador i llibreries d’Objective-C en les quals es basava el sistema operatiu i l’entorn de programació de NeXTstep. Però els ordinadors de NeXT no van aconseguir ser un èxit de ventes, amb la qual cosa es van dedicar exclusivament a la creació de software. Després de la compra de NeXT per part d’Apple el 1996, aquesta última va començar a utilitzar OpenStep en el seu nou sistema operatiu, el MAC OS X. Aquest incloïa Objective-C i l’entorn de programació basat en ell de NeXT’s; anomenat Project Builder, reemplaçat posteriorment pe l’actual XCode (del qual en parlarem més endavant) i la seva eina de disseny gràfic Interface Builder (de la qual també parlarem). Avui en dia són els entorns de desenvolupament de MAC més comuns i l’entorn de desenvolupament de Objective-C en actiu més popular.

La versió actual de l’Objective-C és la 2.0, presentada l’any 2006 a la Conferència Mundial de Desenvolupadors. El MAC OS X v10.5 que va sortir a la venda l’octubre del 2007 ja incloïa el nou compilador per a questa versió, la qual tenia com a novetats respecte la seva predecessora un modern “garbage collector” (o gestor de memòria), una ampliació de la sintaxis existent, una millora del comportament en el temps d’execució de les aplicacions i el suport per a processadors de 64 bits.

6.1.2 CARACTERÍSTIQUES

Les característiques sintàctiques de l’Objective-C són molt similars a les del C, ja que es tracta d’una petita capa sobre el C, estrictament parlant es tracta d’un superconjunt de C. Això fa que sigui possible compilar qualsevol programa en C amb un compilador d’Objective-C. Aquest deriva directament del C i del Smalltalk; heretant la majoria de sintaxi del primer com pre-processament, expressions, declaració de funcions i crides a funcions; mentre que la sintaxi de les propietats OO van ser creades per permetre la missatgeria de tipus Smalltalk.

6.1.2.1 MISSATGERIA

La sintaxi de l’Objective-C ofereix alternatives a algunes de les solucions poc elegants que proporciona la sintaxi del C, però encara més important dona suport a la programació OO. El model de programació OO del Objective-C es basa en l’enviament de missatges a l’objecte concret, això el fa diferent al model “Simula” utilitzat en programació C++. La diferència bàsica de l’Objective-C radica en que no realitza crides als mètodes, si no que els hi envia missatges i aquests els poden emprar o refusar. El tipus de programació OO Smalltalk permet als missatges ser enviats sense implementació i ser creats dinàmicament, però en molts casos s’executen de forma més lenta que els de tipus Simula i són més difícils de depurar.

6.1.2.2 CLASSES I INTERFÍCIES

L'Objective-C requereix la separació en blocs de codi relatius a la interfície i a la implementació de classes. Per convenció es col·loca la interfície en un arxiu de tipus capçalera (header en anglès) normalment de tipus *.h i el cos de la classe (main en anglès) normalment de tipus *.m, els quals són molt similars als de C.

Exemple de interfície:

```
@interface classname : superclassname {
    // instance variables
}
+classMethod1;
+(return_type)classMethod2;
+(return_type)classMethod3:(param1_type)parameter_varName;

-(return_type)instanceMethod1:(param1_type)param1_varName
:(param2_type)param2_varName;
-(return_type)instanceMethod2WithParameter:(param1_type)param1_varName
andOtherParameter:(param2_type)param2_varName;
@end
```

Figura nº 17.

Exemple de implementació:

```
@implementation classname
+classMethod {
    // implementation
}
-instanceMethod {
    // implementation
}
@end
```

Figura nº 18.

6.1.3 OBJECTIVE-C 2.0

Objective-C 2.0 és la major ampliació del llenguatge que suporta l'entorn de programació de Cocoa. Dissenyat com una petit però potent conjunt d'extensions del ANSI-C que suporti el pas de missatges de programació OO (Orientada a Objectes), a més a més és considerat un llenguatge extremadament flexible i dinàmic. El qual et permet treballar a molt alt nivell, construïnt de forma ràpida aplicacions molt sofisticades i mantenint també la capacitat d'accedir a les llibreries de sistema de més baix nivell de la manera més eficient possible. En la figura 19 es pot observar l'estructura de classes:

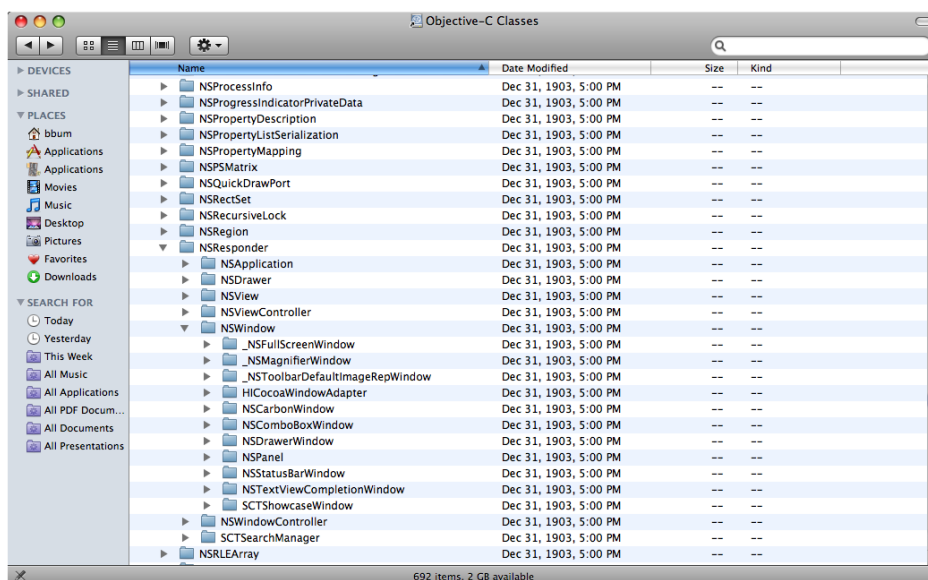


Figura nº 19.

Completament renovat per al sistema operatiu Mac OS X Leopard i les estructures Cocoa de 64-bits, l'Objective-C millora el comportament de les opcions en temps d'execució, proporciona accés a les propietats i a ràpids constructors de sintaxi per iteració, també consta d'un modern i molt millorat sistema de "garbage collector" (introduïrem el concepte a continuació) que permet treure tot l'avantatge possible als últims sistemes multiprocessador. Aquestes noves característiques promocionen el llenguatge al primer nivell del llenguatges moderns OO i et permet escriure millors aplicacions de forma més ràpida i amb menys errors. De la mateixa manera el sistema d'execució continua sent completament compatible amb la versió 1.0 mantenint a les antigues aplicacions actives.

6.1.3.1 GARBAGE COLLECTOR

Els garbage collector són un sistema que implementen els llenguatges per a la gestió de forma automàtica de la memòria. La introducció d'aquesta millora és el canvi més significatiu respecte a les anteriors versions. Com la majoria de llenguatges basats en C, les versions anteriors de Objective-C requerien als programadors per assignar i alliberar la memòria de forma manual. Per ajudar en aquest procés, l'Objective-C proporcionava un comptador de referències per al sistema de gestió de memòria utilitzant les paraules clau "retain" (assignar) i "release" (alliberar). Això simplificava treball, però de totes formes els programadors havien de tenir cura de la gestió de memòria.

Aquest sistema de comptador de la versió 1.0 ha estat substituït, i el nou treballa estretament amb l'Objective-C i totes les llibreries de Cocoa. Això dona la mateixa facilitat de gestió de memòria que altres llenguatges de programació moderns, com el Java. Per oferir el suport a les aplicacions anteriors, el gestor de memòria és implementat com un sistema "opt-in". En el cas de tenir una aplicació ja existent que es vol fer córrer amb un gestor manual de memòria, el programador no ha de fer res.

Anant un pas més endavant, el gestor de l'Objective-C 2.0 és implementat com un gestor conservador, això permet als desenvolupadors tenir una accés total al llenguatge C, preservant la capacitat de l'Objective-C d'integrar el codi i les llibreries de C++. Que significa poder accedir a qualsevol llibreria del

sistema, des del nivell més alt del Core Animation o estructures QuickTime fins al més baix de les crides al sistema. Per suposat quan es treballa amb C o C++ és necessari continuar gestionant la memòria manualment amb les instruccions “malloc” i “free”. En qualsevol cas el resultat és que es pot utilitzar l’eina més eficient i potent per a qualsevol tasca de programació, creant la major part del codi en Objective-C, deixant treballar el gestor de memòria, i quan sigui necessari utilitzar el C o C++ per a crear operacions especials.

6.1.3.2 DECLARACIÓ DE PROPIETATS

Una altra de les característiques importants de l’Objective-C és el suport a propietats declarades; en programació OO les propietats es refereixen a les dades encapsulades mitjançant objectes, als quals s’hi accedeix amb mètodes enlloc d’accedir-hi directament a través de variables. La utilització de variables permet canviar la implementació d’un objecte a posteriori sense afectar la implementació d’altres objectes que utilitzen la mateixa propietat. Per a un millor suport i per animar l’ús de propietats, l’Objective-C 2.0 introdueix una nova sintaxi per a la seva declaració, que permet afegir propietats a les classes creades pel programador amb el mínim codi. En la figura 20 es mostra com declarar una propietat en la capçalera de la classe:

```
@interface Person : NSObject {
    NSString *location;
}
@property NSString *location;
@end
```

Figura nº 20.

Aquest codi indica que la classe “Person” té una propietat anomenada “location” que pot ser llegida i/o assignada. Per a finalitzar la implementació de la propietat és necessari afegir el següent codi a la implementació de la classe:

```
@implementation Person
@synthesize location;
@end
```

Figura nº 21.

6.1.3.3 ENUMERACIÓ RÀPIDA

L’última gran característica de l’Objective-C 2.0 respecte versions anteriors és la ràpida enumeració, la qual permet treballar de forma simple i senzilla amb tots els membres d’una col·lecció, la seva utilització fa el codi més clar i fàcil d’entendre. En l’exemple següent podem observar una iteració sobre un array de strings:

```
NSArray *array =
    [NSArray arrayWithObjects:@"1", @"2", @"3", nil];

for (NSString *string in array) {
    NSLog(@"string is %@", string);
}
```

Figura nº 22.

6.2 XCODE



Xcode^{5 6} és un conjunt d'eines de desenvolupament de software per a Mac OS X (sistema operatiu propietari de Apple) i/o iPhone OS, desenvolupat per Apple. Tot i que també s'hi poden desenvolupar aplicacions de tot tipus capaces de funcionar tant en estructures Intel com PowerPC. La versió 3.0 és la que s'utilitza actualment tot i que ja amb nombroses actualitzacions. L'eina principal del conjunt és l'entorn de desenvolupament integrat (IDE, Integrated Development Environment), també anomenat Xcode. Aquest conjunt està compost també per la majoria de la documentació de desenvolupament d'Apple (Xcode Workspace Guide) i per l'Interface Builder, l'eina utilitzada per a construir les interfícies d'usuari, la qual detallarem més endavant.

El Xcode inclou una versió modificada del programari lliure GNU Compiler Collection (o GCC, que és sistema de compiladors capaç de suportar diferents llenguatges de programació i produït per a GNU), pot treballar amb el codi font de C, C++, Fortran, Objective-C, Objective C++, Java, AppleScript, Python i Ruby. També suporta diferents models de programació com Cocoa (entorn de programació d'aplicacions orientades a objecte d'Apple per a Mac OS X), Carbon (API –interfície de programació d'aplicacions– d'Apple per a Mac OS X) i Java, a més a més de GNU Pascal, Free Pascal, Ada, C#, Pearl, Haskell i D.

En la captura següent es pot observar la pantalla principal del Xcode, tant a la dreta de la imatge com a la part superior hi ha la representació de classes respectant el patró Model-View-Controller (en el que aprofundirem més endavant en la part de Turbogears), amb les seves respectives classes controlador i vista; separades en dos arxius cadascuna i realitzant la funció de capçalera el primer (*.h) i la de contingut el segon (*.m). En aquestes ubicacions també hi ha els directoris dedicats als diferents recursos com les bases de dades, frameworks, imatges, etc... o els arxius necessaris per a qualsevol aplicació com: els executables, informació del projecte o inicialitzadors. La part de baix de la dreta mostra el codi utilitzat en el document seleccionat, el qual és capaç de tractar d'una forma molt eficient amb el seu gestor de texts. Tot el text és classificat segons un codi de colors que indica de que es tracta i també té una funció d'autogeneració, la qual indica les diferents possibilitats existents i estalvia temps al desenvolupador.

⁵ (Apple Team, Xcode, 2008)

⁶ (Anònim, Wikipedia.org, 2008)

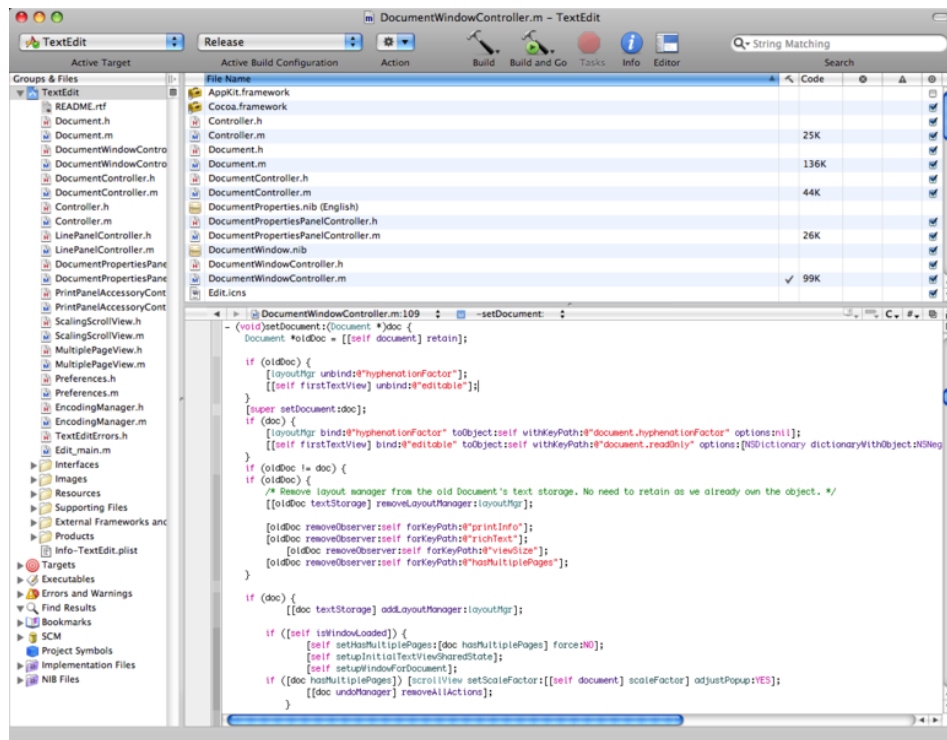
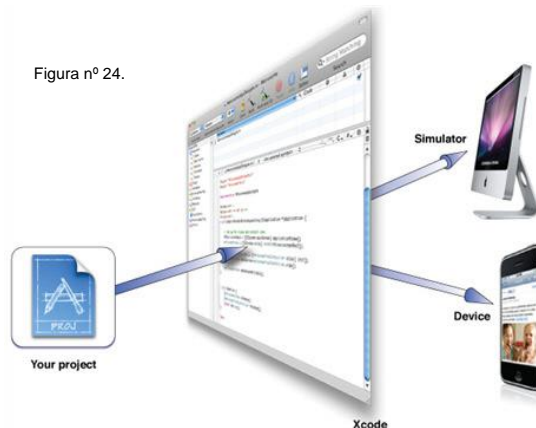


Figura nº 23.

6.2.1 CARACTERÍSTIQUES

Entre les diferents característiques del conjunt Xcode destaca la tecnologia per a distribuir la construcció de codi entre varis ordinadors. Degut a les modificacions realitzades al GCC per Apple, és capaç de crear executables o directoris d'aplicació (en llenguatge Apple, "universal binaries"), que permeten fer córrer a les aplicacions tant en plataformes PowerPC com Intel (x86) i construir-ne en arquitectures de 32 i 64bits. La utilització del SDK (Software Development Kit o conjunt d'eines de desenvolupament de software) específic d'iPhone permet també compilar i depurar aplicacions per a processadors d'arquitectura ARM (Advanced RISC Machine, arquitectura desenvolupada per a sistemes operatius incrustats en dispositius mòbils com l'iPhone) sense el dispositiu. El nom del programari concret és "iPhone Simulator", el qual et permet simular perfectament qualsevol aplicació dissenyada per aquest dispositiu.

Figura nº 24.



6.3 INTERFACE BUILDER



Interface Builder⁷ és una aplicació per a dissenyar i provar interfícies d'usuari (IU) creada per Apple. Els desenvolupadors la poden utilitzar per fer entorns seguint les pautes de Mac OS X i per arrossegar elements de control predefinits des de una paleta fins a una finestra o vista que estiguin configurant. L'Interface Builder treballa estretament amb el Xcode per a facilitar el desenvolupament d'entorns de IU i la lògica de negoci d'una aplicació. És també l'editor gràfic d'Apple per dissenyar les IU per aplicacions del tipus Carbon i Cocoa, permet crear fàcilment programari d'interacció amb l'usuari, dotant als desenvolupadors d'entorns gràfics d'edició per a treballar virtualment cada aspecte del disseny i creant una bona IU que segueixi les pautes de "Aqua". Aqua són les directives marcades per a Apple per a la creació d'entorns de treball entre usuari i maquinari.

6.3.1 FUNCIONAMENT

Interface Builder emmagatzema els recursos de la IU en documents "nib" (xib en el cas de programari per a iPhone). Els documents nib són la representació estàtica del conjunt d'objectes d'interfície utilitzats per a una aplicació, aquests poden ser recuperats de memòria de forma eficient quan sigui necessari, reduint així els temps de desenvolupament i fent més fàcil la localització dels objectes predeterminats per a cada tipus d'aplicació. El desenvolupament amb aquest programari implica normalment:

- Crear nous documents nib Cocoa o Carbon.
- Arrossegar-los fins a una nova finestra o editar una barra de menú.
- Arrossegar tot tipus de components predefinits (botons, barres lliscants, caixes de selecció, etc...) dins de la finestra i configurar els seus atributs (classes Cocoa, comandes d'identificació Carbon, etc...) al panell de informació.
- Els desenvolupadors Cocoa poden crear classes noves, accions i sortides per pantalla (instàncies de variables) i instanciar-les després en un fitxer nib.
- En qualsevol moment el desenvolupador pot utilitzar el "Command-R" per executar la interfície creada en mode de prova. En el mode "Test Interface" pots prémer botons, seleccionar elements de menú, arrossegar finestres, etc..., en general provar la teva IU. Als desenvolupadors d'aplicacions Cocoa també els hi permet de provar les connexions i l'enviament de missatges, així com la possibilitat de testejar les comandes relacionades amb esdeveniments en el domini de la IU.

En la captura de pantalla següent es poden observar les diferents finestres de treball de que disposa el programa per a realitzar aplicacions per a iPhone. En ordre de dreta a esquerra trobem en primer terme la llibreria amb tots els elements predefinits que es poden arrossegar directament sobre la vista. En segon terme trobem l'arxiu on s'emmagatzemen els diferents components que formen la vista, com els objectes que la representen o els elements que hi hàgim pogut arrossegar. En tercer lloc apareix la vista on podem treballar el tema visual arrossegant-hi els diferents elements. Per últim apareix la finestra amb els atributs i característiques que es poden modificar dels components introduïts.

⁷ (Apple Team, Interface Builder, 2008)

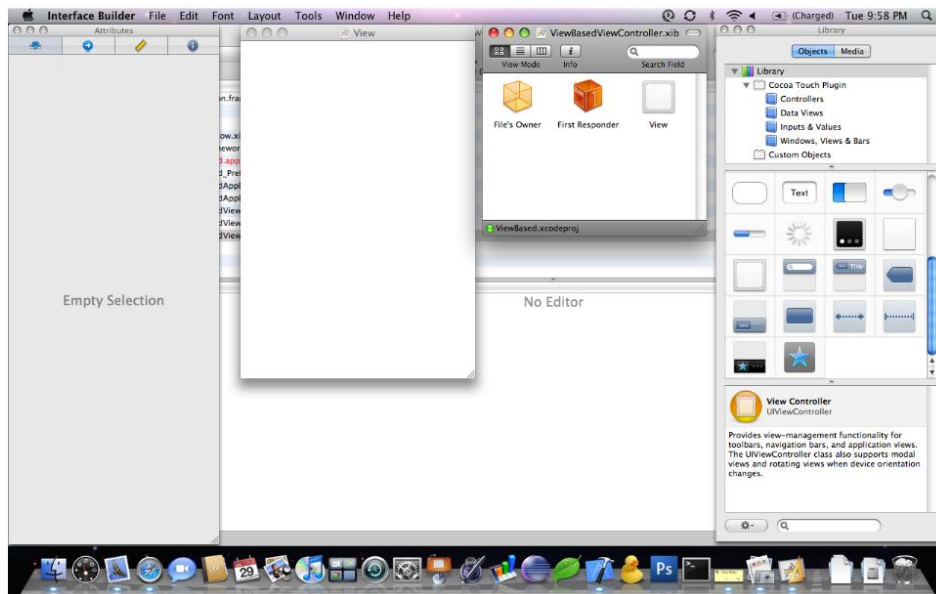


Figura nº 25.

Després de dissenyar la IU en el fitxer nib o xib, es pot fer una d'aquestes dues coses:

- Com a desenvolupador Cocoa, pots guardar l'arxiu nib i les noves classes d'objectes creades i obtenir els fitxers automàticament inserits en el teu projecte Xcode.
- Com a desenvolupador Carbon pots guardar el teu arxiu nib i després recuperar els seus components en el teu programa utilitzant la API IBCarbonRuntime.h.

6.3.2 CARACTERÍSTIQUES

- Ajuda al desenvolupador a dissenyar IU seguint les directrius de Aqua.
- Crea fitxers nib Cocoa i Carbon.
- Simplifica la utilització d'esdeveniments Carbon permeten associar l'identificador de les comandes i/o controls amb els seus controls i/o elements de menú directament des de la vista.
- Dona un àmbit en el procés de desenvolupament d'aplicacions Cocoa que et permet fer subclasses, connectar objectes, establir valors per defecte i ajudar a la aplicació a requerir menys codi.
- Permetre provar la IU en el mode Test Interface sense la necessitat d'escriure codi.
- Corregeix el text de IU.

6.4 SQLITE



SQLite^{8 9 10} és un sistema de gestió de bases de dades (BBDD) SQL independent, sense necessitat de servidor ni configuració i transaccional contingut en una llibreria de C. El codi és de domini públic i

completament gratuït per a qualsevol propòsit tant comercial com privat. Actualment és utilitzat en moltes aplicacions de diferents tipus i finalitats. Està incrustat en un motor de BBDD SQL, a diferència de la majoria d'aquests, no té els processos de servidor separats i és capaç de llegir directament fitxers en discs convencionals. SQLite és una BBDD completa amb múltiples taules, índexs i vistes emmagatzemada en un sol disc de fitxers; el seu format d'arxius pot ser canviat sense problemes entre sistemes de 32 o 64 bits.

SQLite és una llibreria compacta; amb totes les opcions habilitades el seu tamany és inferior a 250 Kb, depenen del compilador i de l'optimització d'aquestes. Si les opcions no estan completament habilitades el tamany de la llibreria pot estar al voltant dels 180Kb, també és capaç de treballar amb memòries de 16Kb, la qual cosa l'ha convertit en un motor molt popular entre els aparells amb poca capacitat com telèfons mòbils, PDAs o reproductors de MP3. Tot i anar millor amb més memòria està preparada per funcionar en entorns de poca memòria degut al compromís adquirit entre el seu ús i la velocitat.

SQLite té una gran reputació de fiabilitat, pràcticament tres quartes parts del codi font estan dedicades a testear i comprovar. També respon correctament als errors d'assignació de memòria i als errors d'entrada/sortida de disc. Les transaccions es realitzen mitjançant ACID (Atomicity, Consistency, Isolation, Durability; conjunt de propietats que garanteixen que les transaccions de BBDD són processades de forma fiable), fins hi tot en cas d'error del sistema o de pèrdues de energia elèctrica. Tot això és verificat per tests automàtics que simulen aquests errors, però tot hi així n' existeixen gran quantitat, els quals són publicats en llistes per a que els usuaris els puguin consultar. El codi base de SQLite és suportat per un equip internacional de desenvolupadors, que s'hi dediquen al 100%. Aquests continuen expandint les capacitats de SQLite, millorant-ne la fiabilitat i mantenint la compatibilitat amb les especificacions d'interfícies publicades, la sintaxi SQL i el format d'arxiu de BBDD. El codi font és completament lliure, però els serveis de suport professional estan també disponibles.

A continuació podem veure la figura 26, eina de desenvolupament SQLite per a Mozilla Firefox que s'ha emprat, en ella es poden observar com es generen les taules, les relacions entre elles i/o les claus primàries. També hi veiem com per la seva part l'eina va generant el codi SQL i la representació abstracta que en fa.

⁸ (Hipp, R. About SQLite, 2008)

⁹ (Hipp, SQLite Documentation, 2008)

¹⁰ (Anònim, Wikipedia.org, 2008)

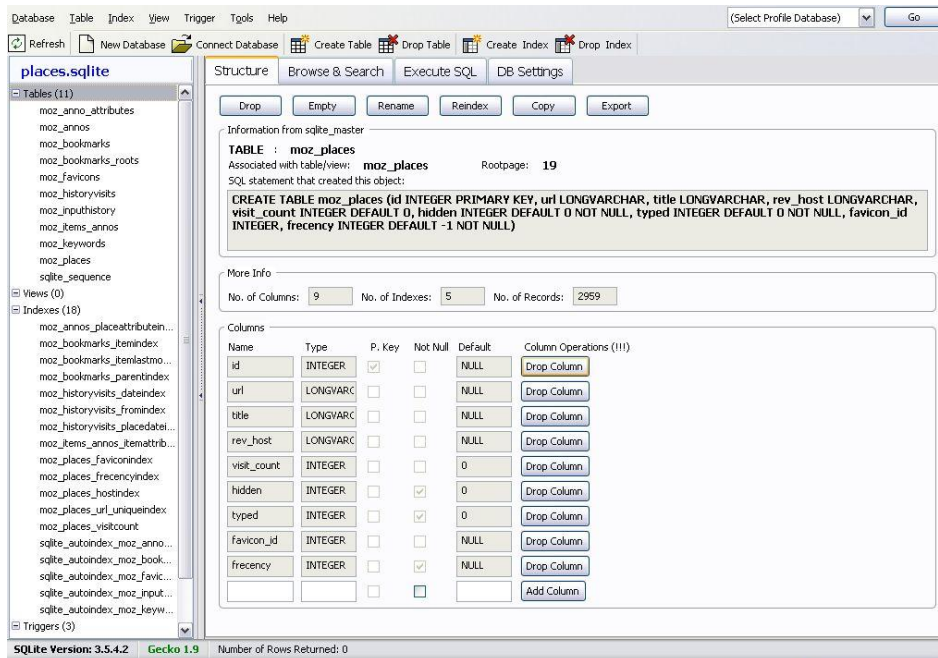


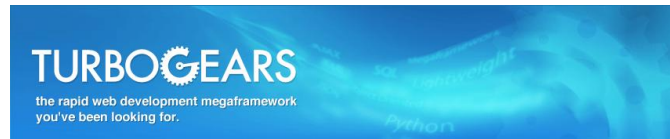
Figura nº 26.

6.4.1 CARACTERÍSTIQUES PRINCIPALS

- Les transaccions són atòmiques, consistents, aïllades i duradores (ACID), fins hi tot en el cas d'errors del sistema o pèrdues d'energia.
- La configuració és mínima.
- Implementa la majoria de SQL92.
- La BBDD és completament emmagatzemada en un sol fitxer sense exclusivitat de plataforma.
- Suporta BBDD de magnitud terabyte i cadenes i/o arxius del tipus "blob" de magnitud gigabyte.
- Poc tamany, menys de 275Kb 100% configurada o menys de 200Kb sense les característiques opcionals.
- Ràpida respecte a populars motors de BBDD client/servidor per a les operacions més comunes.
- Simple, facilitat d'ús de la API.
- Escrita en ANSI-C.
- Codi font ben comentat amb una cobertura superior al 99%.
- Disponibilitat com a fitxer de codi font en ANSI-C, el qual es pot fàcilment exportar a d'altres projectes.
- Polivalència de plataformes: Unix (Linux i Mac OS X), OS/2, Windows (Win 32 i WinCE).
- Codi font de domini públic i utilitzable per a qualsevol propòsit.
- Portable a clients d'interfícies de línies de comandes autònomes (CLI), que poden ser utilitzades per administrar BBDD SQLite.

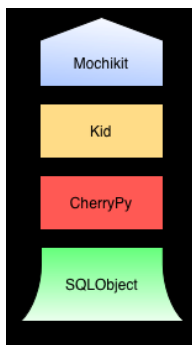
6.5 TURBOGEARS

Fonamentalment s'ha triat Turbogears^{11 12} per a la realització de l'aplicació web que farà les funcions d'actualitzar la BBDD des del costat del servidor degut a l'èxit que han assolit unint una sèrie de components ja existents, en la seva majoria, i dotant-los d'un entorn comú on poder desenvolupar de forma ràpida i eficaç pàgines web. Turbogears és un projecte innovador i bastant recent que intenta facilitar la creació d'una interfície web. En ell es valorat molts aspectes positius com:



- La facilitat d'instal·lació.
- La rapidesa d'aprenentatge dels seus components, ja que són molt intuïtius: com en el cas del codi Python o el SQL.
- L'ús intern d'una estructura de BBDD SQLite, que és el mateix que utilitzen la majoria de dispositius mòbils i també l'iPhone.
- L'ús de plantilles com el Kid, que afavoreixen la modificació de l'entorn d'usuari sense modificar la resta de l'aplicatiu.
- La facilitat per a construir formularis, amb el subministrament de una gran varietat d'elements ja construïts per a la creació de formularis anomenats "widgets"¹³.
- La possibilitat de treballar en una web en marxa i veure els resultats a l'instant, degut al seu sistema de simulació off-line.

Turbogears és la eina escollida per a la realització del back-end (o part del servidor) amb la qual s'actualitzarà la base de dades (o BBDD) quan sigui necessari. La BBDD estarà disponible en un servidor des del qual es descarregarà l'aplicació en el terminal; però, s'ha considerat necessari realitzar tot un aplicatiu perquè una persona externa al projecte, sense coneixements de programació en iPhone o de BBDD, sigui capaç d'actualitzar o modificar aquestes per a la seva posterior visualització en l'aparell mòbil. Turbogears es distribueix a través d'una llicència de "programari lliure" com totes les seves parts. La versió utilitzada de per al desenvolupament ha estat la 1.0, ja que de moment és la versió completa que existeix, la versió 2.0 encara està en fase de desenvolupament.



Com els mateixos creadors de Turbogears la defineixen és "una mega-solució per al ràpid desenvolupament de pàgines web". Aquesta solució inclou una sèrie de components i característiques que la fan realment interessant. Turbogears es basa en el patró de desenvolupament de software Model-View-Controller, el qual també utilitza l'Objective-C i la gran majoria dels llenguatges OO i que descriurem posteriorment. El codi es realitza mitjançant Python, un codi de programació OO, el qual introdueix una sèrie de canvis respecte a llenguatges més tradicionals com el C, Java o el ja esmentat Objective-C; el Python també l'introduïrem més endavant. Altres

¹¹ (Dangoor, Turbogears.com, 2008)

¹² (Dangoor, Turbogears.com, 2008)

¹³ (Anònim, Wikipedia.org, 2009)

components importants del Turbogears són: el MochiKit (una potent llibreria de JavaScript), el Kid (un disseny de sistema de plantilles fàcilment programable), el CherryPy (el qual et permet un sistema de entrada/sortida d'informació web mitjançant l'escriptura de funcions amb Python) i SQLAlchemy (que et dona accés a les teves BBDD mitjançant classes del propi Python). Tot i que per a la BBDD s'ha emprat un altra solució anomenada SQLAlchemy, de la qual en parlarem amb posterioritat.

6.5.1 PATRÓ MODEL-VIEW-CONTROLLER

Model-View-Controller¹⁴ (o MVC) és un patró d'arquitectura de disseny de programari o d'enginyeria del software el qual es basa en la separació en tres elements diferents: les dades de l'aplicació, la interfície de l'usuari i la lògica de control. En el cas que ens ocupa, que és el de la creació de pàgines web, la vista és la pàgina HTML i el codi que proveeix les dades dinàmiques a la pàgina; el model és el sistema de gestió de BBDD i la lògica de l'aplicació i el controlador és l'encarregat de rebre els elements d'entrada des de la vista.

Aquest patró va ser descrit per Trygve Reenskaug el 1979 quan treballava en Smalltalk per l'empresa Xerox. L'èxit d'aquest model resideix en separar la lògica de control de les consideracions de la interfície d'usuari. El resultat és la creació d'una aplicació en la qual es pot modificar fàcilment la interfície de l'usuari o canviar la lògica de control sense que aquests canvis afectin de forma considerable a l'altre part.

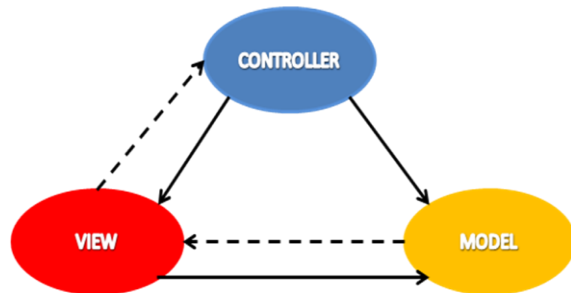


Figura nº 27.

6.5.2 MOCHIKIT

Mochikit^{15 16} és una llibreria de Javascript que inclou el Turbogears, aquesta llibreria agafa conceptes de Python, Objective-C i d'altres llenguatges per a facilitar la feina als desenvolupadors en el procés de creació de programari. A diferència de la majoria de llibreries de JavaScript està escrit amb anglès actual, el que el converteix en un llenguatge molt descriptiu i senzill d'utilitzar. Gràcies a Mochikit es pot incloure directament codi JavaScript a les plantilles i és interpretat pel navegador.



Com podem observar en els exemples següents:

¹⁴ (Anònim, Wikipedia.org, 2008)

¹⁵ (Anònim, Wikipedia.org, 2008)

¹⁶ (MochiMedia, Mochikit.com, 2008)

```

view plain | print | ?
1  myObjectArray = [
2    {"a": 3, "b": 2},
3    {"a": 1, "b": 2}
4  ];
5
6  // sort it by the "a" property, check to see if it matches
7  myObjectArray.sort(keyComparator("a"));
8
9  // get just the "a" values out into an array
10 sortedAValues = map(itemgetter("a"), myObjectArray);

```

Figura nº 28.

En l'exemple anterior crea un objecte del tipus llista i passa els elements ordenats a una altra llista segons el valor de la seva clau.

```

view plain | print | ?
1  var rows = [
2    ["dataA1", "dataA2", "dataA3"],
3    ["dataB1", "dataB2", "dataB3"]
4  ];
5  row_display = function (row) {
6    return TR(null, map(partial(TD, null), row));
7  }
8  var newTable = TABLE({'class': 'prettytable'},
9    THEAD(null,
10     row_display(["head1", "head2", "head3"])),
11    TFOOT(null,
12     row_display(["foot1", "foot2", "foot3"])),
13    TBODY(null,
14     map(row_display, rows)));
15  // put that in your document.createElement and smoke it!
16  swapDOM(oldTable, newTable);

```

Figura nº 29.

En aquest cas crea una funció per a la creació de taules.

6.5.3 KID

Kid¹⁷ és un sistema de plantilles per a realitzar el front-end (o costat de l'usuari) d'una aplicació, està basat en els llenguatges XHTML i XML i forma part de la visualització en el patró model-view-controller. El Kid et permet de forma fàcil i ràpida crear l'aparença d'una pàgina web. Les instruccions per a visualitzar contingut prèviament generat es realitzant amb el llenguatge Python, el qual destaca per la seva senzillesa, facilitat d'ús i d'aprenentatge. Un dels grans avantatges del Kid és la seva capacitat per a generar codi de gran qualitat sense errors en la part de la visualització, això és degut a que està basat en XHTML, cosa que no et permet deixar trossos de codi a l'atzar dins de la plantilla. Un altre dels avantatges és la gran reutilització de les plantilles, amb el que es genera contingut de forma més ràpida i eficient.

```

view plain | print | ?
1  <div py:def="header(title)">
2    <h1>${title}</h1>
3    <p>Here is yet another groovy page of ours.</p>
4  </div>
5
6

```

Figura nº 30.

En aquest exemple d'ús de les plantilles es pot observar com s'introdueixen dades del controlador per a ser enviat tot junt al navegador, el qual a posteriori ho interpretarà segons les especificacions de la plantilla.

¹⁷ (Robert, Kid-templating.org, 2007)

6.5.4 CHERRYPY



CherryPy^{18 19} és una solució web HTTP OO, que utilitza el llenguatge de programació Python. CherryPy et permet generar de forma ràpida i senzilla aplicacions web de la mateixa manera amb la que crearies qualsevol altre programa OO amb Python, embolcallant el protocol HTTP. En la estructura de Turbogears el CherryPy realitza la funció de controlador del patró Model-View-Controller, subministrant una interfície d'enllaç per a les aplicacions web, amb la qual es reben els esdeveniments generats per l'usuari i aquests són retornats a les plantilles.

```
view plain | print | ?
1 import cherrypy
2
3 class MyRoot:
4
5     @cherrypy.expose()
6     def index(self, who="World"):
7         return "Hello, %s!" % (who)
```

Figura nº 31.

Com es pot veure en l'exemple el CherryPy interpreta els arxius de tipus Python que controlaran el contingut que es col·locarà posteriorment en les plantilles.

6.5.5 PYTHON



Python^{20 21 22 23} és un llenguatge dinàmic OO, el qual es pot fer servir en molts tipus diferents de desenvolupament de programari. Ofereix un gran suport per a la integració amb diferents llenguatges i eines, té una gran quantitat de llibreries estàndard i es pot aprendre de forma ràpida i senzilla. És compatible amb Windows, Linux/Unix, Mac OS X, OS/2, Amiga, etc... i també es capaç de treballar amb màquines virtuals Java i .NET. Python va ser publicat el 1991 per Guido van Rossum i és distribuït i gestionat com a programari lliure per la fundació sense ànim de lucre Python Software Foundation, la qual s'encarrega de mantenir la definició estàndard del llenguatge. En aquesta aplicació s'ha utilitzat la versió 2.5 del programari, ja que és la recomanada per a funcionar amb la versió 1.0 de Turbogears.

Aquest codi és comparat amb d'altres com: Tcl, Pearl, Ruby, Scheme o Java. Però hi ha una sèrie de característiques que el fan diferent de tots aquests, i són les següents:

- Molt clar i amb una sintaxi fàcilment llegible.
- Gran capacitat d'introspecció.
- Molt intuïtiu en el referent a OO.
- Naturalitat en les expressions procedimentals.

¹⁸ (Anònim, Wikipedia.org, 2008)

¹⁹ (Edgewall Software Team, CehhryPy.org, 2008)

²⁰ (Python Software Foundation, Python.org, 2008)

²¹ (Python Software Foundation, Python.org About, 2008)

²² (Python Software Foundation, Python.org Documentation, 2008)

²³ (Anònim, Wikipedia.org, 2008)

- Totalment modular i amb suport d'herència de paquets.
- Tipus de dades dinàmiques de molt alt nivell.
- Grans llibreries estàndard i una tercera part dels mòduls dedicats a la virtualització de tasques.
- Extensions i mòduls fàcilment creables amb C, C++, Java o .NET.
- Capacitat de ser incrustat sense aplicacions com una interfície d'escriptura.

A continuació introduïrem una sèrie d'exemples en codi Python:

```
>>> def iterquad ():
...     for i in range(5):
...         yield (i*i)
...
>>> for j in iterquad():
...     print j
...
0
1
4
9
16
```

Figura nº 32.

En aquest exemple senzill es pot observar com es defineixen funcions en Python i la seva forma diferent d'utilitzar el bucle "for". Cal apreciar, a diferència d'altres llenguatges que no fan falta les claus per a indicar on comença i/o acaba el codi de les funcions. Els caràcters de color vermell són els generats per les instruccions.

```
>>> class Student:
...     def __init__ (self, name, age, gender):
...         self.name = name
...         self.age = age
...         self.gender = gender
...
>>> Sue = Student("Susan Miller", 20, "f")
>>> print Sue
<__main__.Student instance at 0x81a96cc>
>>> print Sue.age
20
```

Figura nº 33.

A l'exemple podem veure com es defineixen noves classes en Python i com funciona OO en aquest codi. La funció "__init__" és l'estàndard en totes les classes i la qual podem sobreescrivre per a modificar-ne el seu comportament. També cal destacar com es generen les variables, les quals no és necessari declarar i obtenen el seu tipus dinàmicament al passa'ls-hi contingut.

```
>>> from types import *
>>>
>>> def what (x):
...     if type(x) == IntType:
...         print "This is an int."
...     else:
...         print "This is something else."
...
>>> what(4)
This is an int.
>>>
>>> what("4")
This is something else.
```

Figura nº 34.

En aquest darrer exemple es pot comprovar més clarament com s'assigna de forma dinàmica el tipus de les variables, la definició de funcions i com s'importen les diferents llibreries mitjançant la instrucció "import".

6.5.6 SQLALCHEMY



SQLAlchemy^{24 25 26 27} és l'eina de desenvolupament SQL de Python i un mapejador d'objectes relacionals (ORM), el qual ofereix als desenvolupadors tota la potència i flexibilitat del SQL. SQLAlchemy et dona accés a la BBDD de forma senzilla i pràctica, sense haver de treballar directament amb el SQL. Permet desenvolupar amb comoditat les taules, convertint les seqüències SQL en operacions amb objectes. Actualment és dona un altre suport per a BBDD en Turbogears anomenat SQLAlchemyObject, però s'ha decidit treballar amb SQLAlchemy per varies raons: és un ORM més potent que el ja esmentat SQLAlchemyObject, en futures versions aquest serà el ORM en Turbogears i és molt senzill d'utilitzar. La versió emprada per al desenvolupament es la 0.5. A continuació veurem un parell d'exemples del seu funcionament:

```
1. from sqlalchemy import *
2. from sqlalchemy.ext.declarative import declarative_base
3. from sqlalchemy.orm import relation, backref, sessionmaker
4.
5. Base = declarative_base()
6. metadata = Base.metadata
7.
8. class User(Base):
9.     __tablename__ = 'users'
10.
11.     id = Column(Integer, primary_key=True)
12.     name = Column(String(20))
13.     fullname = Column(String(50))
14.
15.     def __init__(self, name, fullname):
16.         self.name = name
17.         self.fullname = fullname
18.
19.     def __repr__(self):
20.         return "<User('%s','%s')>" % (self.name, self.fullname)
21.
22. db = create_engine('sqlite:///database.sqlite', echo=False)
23. metadata.create_all(db)
```

Figura nº 35.

En l'exemple anterior es pot comprovar com el SQLAlchemy et permet generar una BBDD amb les seves taules i els camps de cada taula, assignant-li els tipus, claus primàries, etc...

²⁴ (SQLAlchemy Team, The Python SQL Toolkit and Object Relational Mapper, 2008)

²⁵ (SQLAlchemy Team, SQLAlchemy 0.5.3 Documentation, 2008)

²⁶ (SQLAlchemy Team, Key Features of SQLAlchemy, 2008)

²⁷ (Anònim, Wikipedia.org, 2008)

```

1. class Address(Base):
2.     __tablename__ = 'addresses'
3.
4.     id = Column(Integer, primary_key=True)
5.     email_address = Column(String, nullable=False)
6.     user_id = Column(Integer, ForeignKey('users.id'))
7.
8.     user = relation(User, backref=backref('addresses',
9.         order_by=id, cascade='all, delete-orphan',
10.         passive_deletes=False))
11.
12.     def __init__(self, email_address):
13.         self.email_address = email_address
14.
15.     def __repr__(self):
16.         return "<Address('%s')>" % self.email_address

```

Figura nº 36.

En aquest cas el codi ens dona un exemple de com realitzar les relacions entre les diferents taules.

7. IMPLEMENTACIÓ

Un cop finalitzada la fase inicial d'aprenentatge del llenguatge Objective-C i de les eines de treball per a programació en iPhone com Xcode, Interface Builder i iPhone Simulator; i també desenvolupat el disseny de les diferents parts del sistema, el projecte entra en la fase de implementació i desenvolupament del mateix. Aquest capítol vol ser una aproximació a les diferents parts que componen aquest desenvolupament; comentant les diferents problemàtiques aparegudes, com s'han resolt i els passos necessaris realitzats per a la finalització de tot el conjunt.

7.1 IPHONE

Per a començar el projecte és necessari arrancar el programari Xcode, descrit en el capítol anterior, i realitzar un nou projecte. Al realitzar aquesta acció s'obre un assistent on s'ha de triar el tipus de projecte que es desitja implementar, en el nostre cas "Window-Based Application". Això et proporciona un projecte amb els diferents elements bàsics per a començar el desenvolupament, amb la seva classe "Delegate" i un fitxer del tipus *.xib (els quals representen les vistes), amb el qual es pot començar a treballar la visualització. Per visualitzar l'aplicació generada per a l'iPhone s'ha realitzat la figura Nº 37.

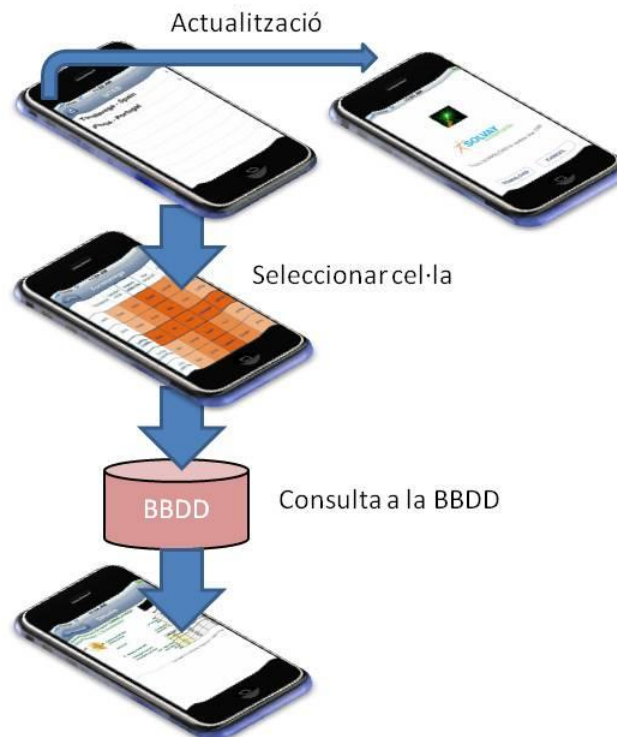


Figura nº 37.

7.1.1 DESCRIPCIÓ DEL CODI

La següent figura representa el diagrama de classes resultant de la realització del programari:

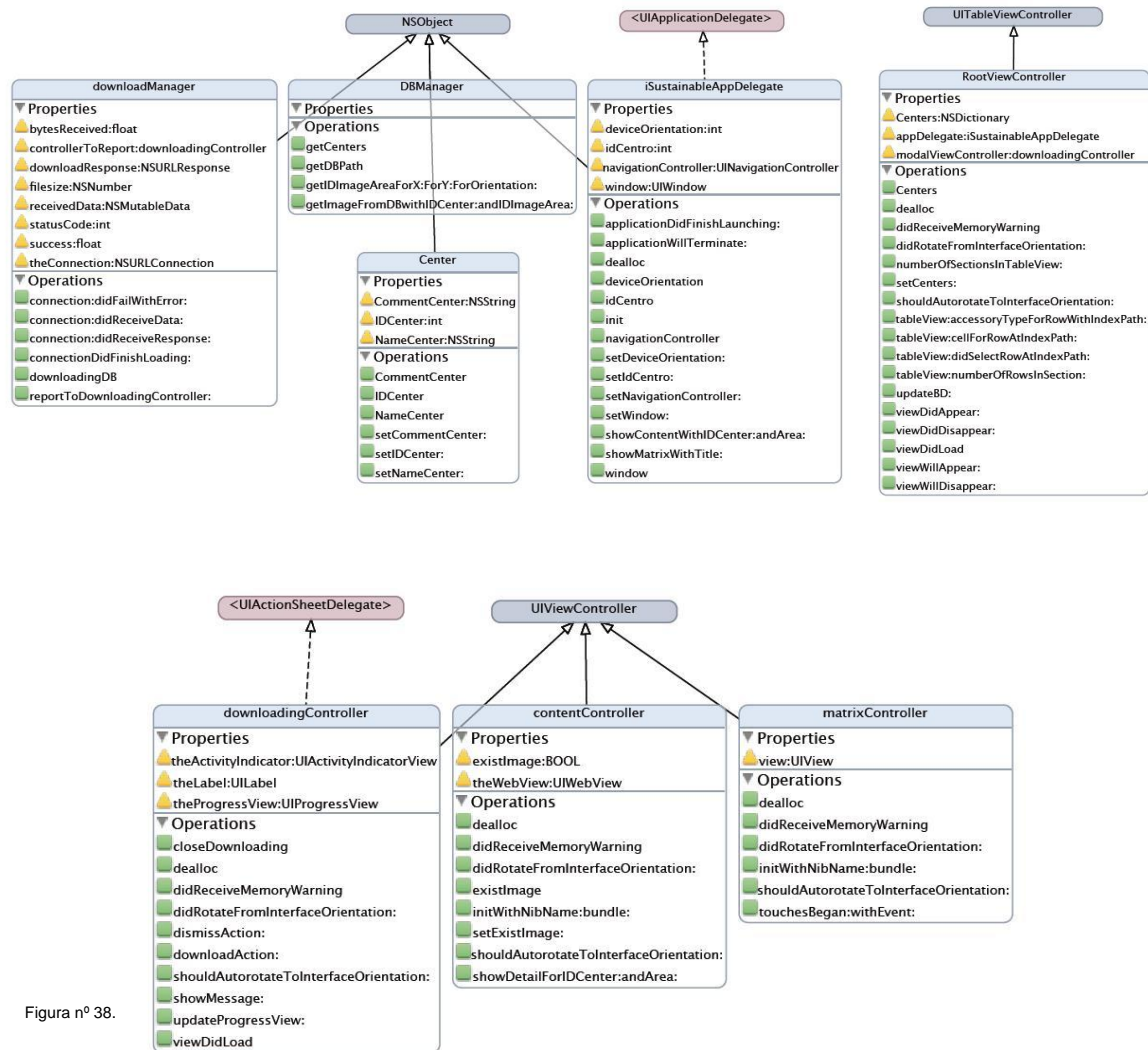


Figura nº 38.

A continuació introduïrem les diferents classes descrites en la figura anterior, les relacions entre elles i els seus mètodes:

- **NSObject**: Classe arrel del Objective-C per a la programació en iPhone OS 2.0 o superior, de la qual hereten la gran majoria de classes. A través de NSObject els objectes hereten la interfície bàsica en temps d'execució del sistema i el comportament d'objectes d'Objective-C.
- ◆ **Center**: Classe heretant de NSObject, la qual representa un objecte amb la informació d'un registre de la taula Center.
 - **NameCenter**: Mètode de la classe Center, el qual retorna el nom del centre.

- setNameCenter: Mètode de la classe Center, el qual assigna el valor de la propietat NameCenter.
 - CommentCenter: Mètode de la classe Center, el qual retorna el país del centre.
 - setCommentCenter: Mètode de la classe Center, el qual assigna el valor de la propietat CommentCenter.
- ◆ DBManager: Classe heretant de NSObject, la qual agrupa les diferents interaccions amb la BBDD.
- getIdImageAreaForX: ForY: ForOrientation: Mètode de la classe DBManager, el qual realitza una consulta a la BBDD i retorna un enter amb la id de la cel·la de la matriu on ha polsat l'usuari.
 - getCenters: Mètode de la classe DBManager, el qual realitza una consulta a la BBDD i retorna un diccionari amb tots els objectes Center de la taula.
 - getImageFromDBwithIDCenter: andImageArea: Mètode de la classe DBManager, el qual realitza una consulta a la BBDD i retorna la imatge emmagatzemada.
 - getDBPath: Mètode de la classe DBManager, el qual comprova si existeix una BBDD descarregada i en retorna la ubicació, en cas contrari utilitza la BBDD interna al projecte.
- ◆ downloadManager: Classe heretant de NSObject, la qual controla el procés d'actualització de la BBDD i la seva connexió amb el servidor.
- downloadingDB: Mètode de la classe downloadManager, el qual realitza la connexió amb el servidor.
 - connection: didReceiveResponse: Mètode de la classe downloadManager, el qual comprova la resposta del servidor i retorna un missatge d'error si és necessari o el tamany del fitxer de la BBDD.
 - connection: didReceiveData: Mètode de la classe downloadManager, el qual copia el fitxer descarregat al directori "Documents" del terminal.
 - connectionDidFinishLoading: Mètode de la classe downloadManager, el qual espera la resposta del servidor i en el cas de ser afirmativa treu un missatge per pantalla.
 - connection: didFailWithError: Mètode de la classe downloadManager, el qual treu un missatge d'error per pantalla en el cas d'haver-hi algun tipus de problema amb la connexió.
 - reportToDownloadingController: Mètode de la classe downloadManager, el qual assigna un objecte del tipus DownloadingController a un altre objecte.
- UIApplicationDelegate: Protocol d'Objective-C que hereta de NSObject. El qual declara els mètodes que son implementats pel "delegate" de tipus "singleton" de l'objecte UIApplication. Amb la implementació d'aquests mètodes, el delegate és capaç de respondre a la iniciació/tancament de la aplicació, a les alertes de poca memòria disponible, a l'obertura de recursos URL, als canvis d'orientació en la barra d'estat i a d'altres esdeveniments del sistema.
- ◆ iSustainableAppDelegate: Classe heretant de NSObject i del protocol UIApplicationDelegate, que realitza les funcions de classe principal de l'aplicació.

- showMatrixWithTitle: Mètode de la classe iSustainableAppDelegate, el qual inicialitza la primera vista de la aplicació (Matrix.xib).
 - showContentWithIDCenter: andArea: Mètode de la classe iSustainableAppDelegate, el qual inicialitza la vista Content.xib, la qual mostra les transparències.
 - applicattionDidFinishLaunching: Mètode de la classe iSustainableAppDelegate, el qual carrega la primera vista amb la taula i els centres.
- UITableViewController: Classe d'Objective-C que hereta de UIScrollView, UIView, UIResponder i NSObject. La qual et permet mostrar i editar llistes d'informació en forma de taula.
- ◆ RootViewController: Classe heretant de UITableViewController, que realitza la funció de controlador arrel i de la taula inicial.
 - viewDidLoad: Mètode de la classe RootViewController, el qual s'executa una vegada carregada la vista, assigna l'acció al botó actualitzar i inicialitza la view downloading.xib.
 - updateDB: Mètode de la classe RootViewController, el qual carrega la vista d'actualització presentModelViewController.
 - numberOfSectionsInTableView: Mètode de la classe RootViewController, el qual retorna el número de seccions de la taula.
 - tableView: numberOfRowsInSection: Mètode de la classe RootViewController, el qual retorna el nombre de centres emmagatzemats a la BBDD.
 - tableView: cellForRowAtIndexPath: Mètode de la classe RootViewController, el qual inicialitza les cel·les de la taula, carrega el contingut de les cel·les i s'executa tantes vegades com número de centres.
 - didRotateFromInterfaceOrientation: Mètode de la classe RootViewController, el qual permet saber si el dispositiu es troba en posició horitzontal o vertical i guardar-la.
 - tableView: didSelectRowAtIndexPath: Mètode de la classe RootViewController, el qual s'executa al seleccionar una de les cel·les de la llista i crida la matriu amb el contingut del centre seleccionat.
 - tableView: accessoryTypeForRowWithIndexPath: Mètode de la classe RootViewController, el qual afegeix les fletxes en l'interior de les cel·les.
 - shouldAutorotateToInterfaceOrientation: Mètode de la classe RootViewController, el qual et permet girar la vista d'acord amb la posició del terminal.
- UIViewController: Classe d'Objective-C que hereta de UIResponder i NSObject. La qual proporciona el control i accés sobre la vista. S'utilitzen les instàncies d'aquesta i les seves subclasses per a controlar les diferents barres o elements que es poden incloure en una vista. També suporta les vistes especials i de rotació quan canvia l'orientació del terminal.
- ◆ matrixController: Classe heretant de UIViewController, que realitza la funció de controlador de la vista de la matriu.
 - initWithNibName: bundle: Mètode de la classe matrixController, el qual inicialitza la visualització del fitxer *.xib.

- touchesBegan: withEvent: Mètode de la classe `matrixController`, el qual controla els esdeveniments que es produeixen quan l'usuari pitja en algun punt de la pantalla, obtenint la cel·la de la matriu que ha sigut pitjada.
- ◆ contentController: Classe heretant de `UIViewController`, que realitza la funció de controlador de la vista de continguts.
 - showDetailsForIDCenter: andArea: Mètode de la classe `matrixController`, el qual carrega la imatge emmagatzemada a la BBDD segons el centre i la area de la matriu seleccionada.
- UIActionSheetDelegate: Protocol d'Objective-C que hereta de `NSObject`. El qual defineix els mètodes "delegate" que un objecte del tipus `UIActionSheet` ha d'implementar. El "delegate" implementa les accions dels botons i alguns altres comportaments programables. La majoria d'aquests mètodes són opcionals.
- ◆ downloadingController: Classe heretant de `UIViewController` i de `UIActionSheetDelegate`, que realitza la funció de controlador de la vista de l'actualització.
 - dismissAction: Mètode de la classe `downloadingController`, el qual cancel·la la descàrrega de d'una nova BBDD en el cas d'haver-la iniciat i torna al menú principal o només torna al menú principal.
 - downloadAction: Mètode de la classe `downloadingController`, el qual inicia la descàrrega de la BBDD al pitjar al botó "Download".
 - showMessage: Mètode de la classe `downloadingController`, el qual mostra per pantalla els diferents missatges d'estat de la descàrrega de la BBDD.
 - closeDownloading: Mètode de la classe `downloadingController`, el qual tanca la vista d'actualització.

7.1.2 EXEMPLES D'ALGORITMES

A continuació unes quantes línies de codi d'exemple de com s'han resolt diferents problemàtiques. En el primer exemple es pot observar el mètode `getDBPath` de la classe `DBManager`. Amb el qual s'ha resolt la problemàtica de tenir dues BBDD diferents, una d'elles interna en el projecte i l'altra en un directori del terminal. Retornant la ubicació on es troba la BBDD:

```

-(NSString *)getDBPath{

    NSArray *pathList = NSSearchPathForDirectoriesInDomains(NSDocumentDirectory, NSUserDomainMask, YES);
    NSString *documentPath = [[pathList objectAtIndex:0] stringByAppendingPathComponent:@"sustainable2.sqlite"];
    NSLog(documentPath);

    NSFileManager *fileManager = [NSFileManager defaultManager];

    if ( ! [fileManager fileExistsAtPath:documentPath] ) {

        NSString *defaultDocumentPath = [[NSBundle mainBundle] pathForResource:@"sustainable2" ofType:@"sqlite"];

        documentPath = defaultDocumentPath;
    }
}

```

```

    return documentPath;
}

```

Figura nº 39.

El segon exemple, també de la classe DBManager il·lustra com es realitza una consulta a la BBDD. Aquest mètode retorna el valor de la identitat de la cel·la de la matriu pitjada:

```

-(int)getIdImageAreaForX:(int)coordinateX ForY:(int)coordinateY ForOrientation:(int)orientation{

    NSString *query1 = [NSString stringWithFormat:@"SELECT IDImageArea FROM ImageArea WHERE
ImagePosition=%i",orientation];
    NSString *query2 = [NSString stringWithFormat:@" AND x1<=%i",coordinateX];
    NSString *query3 = [NSString stringWithFormat:@" AND x2>=%i",coordinateX];
    NSString *query4 = [NSString stringWithFormat:@" AND y1<=%i",coordinateY];
    NSString *query5 = [NSString stringWithFormat:@" AND y2>=%i",coordinateY];
    NSString *queryA = [query1 stringByAppendingString:query2];
    NSString *queryB = [queryA stringByAppendingString:query3];
    NSString *queryC = [queryB stringByAppendingString:query4];
    NSString *query = [queryC stringByAppendingString:query5];

    static sqlite3 *db;
    int msg;
    sqlite3_stmt *statement;
    const char *next;
    const char *pathCStr;

    pathCStr = [[self getDBPath] UTF8String];
    assert(pathCStr != NULL);

    msg = sqlite3_open(pathCStr,&db);
    msg = sqlite3_prepare(db, [query UTF8String], [query length], &statement, &next);
    int idImageArea = 0;

    while(sqlite3_step(statement)==SQLITE_ROW){

        idImageArea = [[NSString stringWithUTF8String:(char *)sqlite3_column_text(statement, 0)]intValue];
    }

    return idImageArea;
}

```

Figura nº 40.

7.2 APLICACIÓ D'ACTUALITZACIÓ

Per a realitzar la aplicació d'actualització s'ha utilitzat l'eina Turbogears. Per a treballar amb aquesta eina és necessari tenir instal·lat Python a l'ordinador i descarregar el fitxer tgsetup.py de la web de Turbogears. Un cop fet això és tant fàcil com cridar aquest fitxer des de la consola i Turbogears queda completament instal·lat. Per a treballar-hi és necessari disposar d'un IDE (Entorn Integrat de Desenvolupament), en aquest s'ha triat Notepad++; qualsevol navegador d'Internet (Mozilla Firefox, Internet Explorer, Safari, etc...) i de la consola de comandes de Windows en aquest cas.

Començar un nou projecte bàsic es redueix a escriure tres línies a la consola de comandes: "tg-admin quickstart" per a crear el projecte, "cd nom_del_projecte" i "python start-nom_del_projecte.py" per a llançar la aplicació. Un cop fet això anant al navegador i introduint la següent direcció és visualitza el projecte: <http://localhost:8080/>. És d'aquesta manera, ja que Turbogears simula el funcionament d'un servidor de forma local, i podem desenvolupar tot el projecte sense la necessitat disposar-ne d'un.

Al crear el projecte, Turbogears genera un directori amb una sèrie d'elements imprescindibles per al seu correcte funcionament. Apart del fitxer anomenat anteriorment "start-nom_del_projecte.py" i d'altres elements, genera un altre directori, dintre del qual trobem el fitxer "controllers.py" on definirem els diferents controladors; el directori "templates" on hi ha les plantilles de les diferents pàgines i el directori "static" on s'emmagatzemen les imatges, fitxers *.css, etc... A continuació aprofundirem en tots i cadascun d'ells.

7.2.1 CONTROLLERS

Controllers és el fitxer principal de qualsevol aplicació generada amb Turbogears, en ell es troben totes les classes i mètodes de la aplicació. S'hi defineixen les diferents funcionalitats i el comportament de totes les classes i objectes. Tots els mètodes es poden cridar des del navegador, rebre valors per paràmetre a través de la URL i estar o no lligats a un "template" (o plantilla en anglès) amb el que interactuen per a mostrar un contingut totalment programable. El llenguatge utilitzat per a la programació dels diferents mètodes és Python.

Primer de tot s'importen els diferents mòduls, sencers o alguna de les seves classes o funcions, com en els exemples següents:

```
import turbogears as tg
from cherrypy import request, response
```

Figura nº 41.

A continuació es defineix la classe "Root", on hi declararem tots els mètodes:

- index: controla el template principal.
- login: Controla la identificació dels usuaris que volen accedir a l'aplicació i l'accés als seus mètodes. Generat de forma automàtica al crear un projecte amb identificació.
- logout: Realitzat per a sortir de l'aplicació. També es genera de forma automàtica al realitzar un projecte amb identificació.
- add_delete_center: Controla el formulari inicial d'edició dels centres, en el qual hi utilitzem "widgets" per a la creació d'elements com botons, camps de text o llistes de selecció d'elements, definits en l'interior de la classe "form_edit". Retorna un objecte de tipus diccionari amb els diferents elements del formulari i la direcció d'URL que visitarà quan l'usuari pitgi el botó per defecte del formulari per a enviar la informació. També realitza una crida al mètode "getCenters" per a omplir la llista de selecció d'elements amb els diferents centres ja existents a la BBDD.
- getCenters: Realitza una consulta a la BBDD per a obtenir els diferents centres i construeix l'estructura (identitat,centre) necessària per a la creació de la llista de selecció d'elements.
- addCenter: Insereix a la taula "Center" de la BBDD un nou registre amb la seva identitat i els camps "Name" i "Comment" amb la informació introduïda per l'usuari i crida de nou a "add_delete_center". El mètode és cridat des del botó "Add" del template add_delete_center.
- deleteCenter: Elimina de la taula "Center" de la BBDD el registre seleccionat per l'usuari i crida de nou a "add_delete_center", a més a més també elimina tot el contingut relacionat

amb el centre a les taules “Image” i “Content”. El mètode és cridat des del botó “Delete” del template add_delete_center.

- publish: Realitza una còpia de la BBDD al servidor utilitzat per a la descàrrega des del terminal iPhone i crida de nou a “add_delete_center”. El mètode és cridat des del botó “Save changes” del template add_delete_center.
- edit: Està lligat al template de la matriu mapejada. Realitza una consulta a la BBDD per mostrar el nom i el país del centre seleccionat. Retorna un objecte diccionari amb els resultats de la consulta per a mostrar-los per pantalla.
- modify: Està lligat al template que mostra la imatge que en aquells moments està emmagatzemada a la BBDD pel centre i la cel·la de la matriu seleccionats. Realitza una consulta sobre la BBDD per a obtenir la imatge i la copia al directori “static/images/test.jpg” del servidor per a la posterior visualització. Si no existeix cap imatge a la BBDD elimina el fitxer “test.jpg” i mostra un espai en blanc. També extreu els elements del formulari “form_modify_DB” declarat al fitxer forms.py.
- deleteImage: Elimina de la taula “Image” de la BBDD el registre seleccionat per l’usuari i crida de nou a “modify”, també elimina tots els registres relacionats amb la imatge a la taula “Content”. El mètode és cridat des del botó “Delete Image” del template modify. Un cop realitzat això torna a la pàgina modify.
- DBModified: Insereix la imatge seleccionada per l’usuari i/o els comentaris realitzats a la taula Image i també genera els registres necessaris a la taula Content. En el cas de que la imatge ja existeixi realitza una modificació sobre els registres ja creats. Un cop fet això torna a la pàgina modify. El mètode és cridat des de el botó “Submit” (nom segons l’idioma del navegador) del template.

A continuació unes quantes línies de codi d’exemple de com s’han resolt diferents problemàtiques. En el primer exemple es pot observar el mètode addCenter. Amb el qual s’ha resolt la problemàtica d’introduir un nou registre a la taula Center de la BBDD:

```
@expose()
@identity.require(identity.in_group("admin"))
def addCenter(self, forward_url=None, **param):

    if('center' in param):
        engine = create_engine(self.db_path,echo=False)
        metadata = MetaData()
        metadata.bind = engine
        centers = Table('center', metadata, autoload=True)
        contents = Table('content', metadata, autoload=True)
        ins = centers.insert().values(Name=param["center"], Comment=param["country"])
        conn = engine.connect()
        result = conn.execute(ins)

        redirect("/add_delete_center")

    else:
        redirect('/')

    return (param["center"])
```

Figura nº 42.

El següent mètode, deleteCenter, resol la problemàtica d’eliminar un Centre i tots els seus registres associats:

```

@expose()
@identity.require(identity.in_group("admin"))
def deleteCenter(self, forward_url=None, **param):

    if('IDCenter' in param):
        if (int(param['IDCenter'])>=0):
            con = sqlite.connect(self.db_path1)
            cur = con.cursor()
            cur.execute('DELETE FROM Center WHERE IDCenter=' + param['IDCenter'])
            cur.execute('DELETE FROM Image WHERE IDImage IN (SELECT Image.IDImage FROM Image JOIN Content ON
Image.IDImage=Content.IDImage WHERE IDCenter=' + param['IDCenter'] + ')')
            cur.execute('DELETE FROM Content WHERE IDCenter=' + param['IDCenter'])
            cur.execute('VACUUM')
            con.commit()
            con.close()
            redirect("/add_delete_center")

        else:
            flash("You have to select one center to delete it!")
            redirect("/add_delete_center")

    else:
        redirect('/')

    return (param["delete"])

```

Figura nº 43.

7.2.2 FORMS

El fitxer forms.py és un fitxer generat per a la declaració dels elements pertanyents a un formulari. Aquests elements són widgets, un widget és una petita aplicació o programa, normalment presentat en petits fitxers, els quals són executats per un motor de widgets. Els seus objectius són donar un fàcil accés a funcions utilitzades amb molta freqüència i proveir d'informacions visuals. En el cas de Turbogears es tracta d'una forma senzilla i potent de combinar codi HTML, CSS i JavaScript en components reutilitzables. Aquests components poden ser botons, llistes desplegable, quadres de text, etc...

Les següents línies de codi són un exemple del que s'ha fet en aquest fitxer, això s'ha realitzat per a tenir el codi més localitzat, de forma més clara i concisa. Un cop implementat en el forms.py és tant senzill com realitzar un "import" en el controllers. Aquesta classe és utilitzada pel mètode modify:

```

class form_modify_DB(widgets.WidgetsList):

    navigate = widgets.FileField("choose", label="")
    comment = widgets.TextArea("comment", label="", rows=5, cols=40)

```

Figura nº 44.

7.2.3 DEVDATA.SQLITE

BBDD generada pel propi Turbogears quan es realitza un projecte amb identificació. La qual s'utilitza per emmagatzemar els diferents usuaris que es vulguin crear per a restringir o permetre l'accés a l'aplicació. També és capaç d'emmagatzemar grups, amb els quals es poden assignar diferents perfils i emmagatzemar les diferents visites guardant en aquest cas l'usuari que hi ha accedit.

7.2.4 TEMPLATES

Els templates en Turbogears són fitxers del tipus *.kid, kid és el sistema de plantilles de fàcil utilització implementat per Turbogears. Aquestes plantilles s'implementen en llenguatge HTML, tot i que també interpreten llenguatge JavaScript. Treballen amb concordança amb els mètodes creats al controllers.py, aquests es lliguen mitjançant la instrucció següent:

"@expose(template="nom_del_projecte.templates.nom_del_template")". Aquesta instrucció es col·loca a la capçalera del mètode. La comunicació entre la plantilla i el mètode es realitza mitjançant l'enviament d'objectes de tipus diccionari per part del mètode, els quals els rep i interpreta la plantilla amb la instrucció d'HTML següent: "\${nom_de_l'objecte()}", l'objecte diccionari són els valors de retorn del mètode.

L'exemple següent és el mètode convertText(), que forma part de la plantilla add_delete_center. El qual comprova si el centre que volem afegir existeix o no, en el cas que no existeixi crida al mètode add_center per afegir-lo:

```
function convertText(){

    var len=document.form.form_centersList.length;
    var existeCentro=0;

    if(!document.form.form_center.value.toUpperCase() && existeCentro != 1){

        alert("You have to enter a center!")
        existeCentro = 1;

        return;
    }

    for(a=0;a<=len-1;a++){
        alert(document.form.form_centersList.options[a].text.toUpperCase());
        alert(document.form.form_center.value.toUpperCase());

        if(document.form.form_centersList.options[a].text.toUpperCase() == document.form.form_center.value.toUpperCase()){

            alert("Center already exists!");
            existeCentro = 1;
            return;
        }
    }

    if(existeCentro==0){

        window.location='http://localhost:8080/addCenter?center=' + document.form.form_center.value + '&country=' +
        document.form.form_country.value;
        existeCentro = 1;
    }
}
```

Figura nº 45.

El següent mètode deleteImg(), forma part de la plantilla modify. És la forma que s'ha utilitzat per a cridar un altre mètode del controllers.py pitjant en un botó, també s'ha utilitzat en d'altres plantilles:

```
function deleteImg(idImage,idCenter,idArea1,idArea2){

    window.location='http://localhost:8080/deleteImage?IDImage=' + idImage + '&IDCenter=' + idCenter + '&IDImageArea1=' +
    idArea1 + '&IDImageArea2=' + idArea2;

}
```

Figura nº 46.

7.2.5 CSS, IMAGES I FILES

En Turbogears també existeix un full d'estil, el qual es programa en llenguatge css. Aquest full permet crear estils pre-configurats que després es poden emprar en les diferents plantilles. Sense la necessitat de definir-los cada vegada. També existeix un directori anomenat Images s'emmagatzemen les diferents imatges per defecte. En el directori files s'emmagatzemen els diferents arxius que es poden descarregar directament des de el lloc web: com la BBDD, el perfil d'usuari o la mateixa aplicació.

7.3 FASE DE PROVES I DISTRIBUCIÓ

La fase de proves es va iniciar tant en l'aplicació per iPhone com en la aplicació web d'actualització quan es va obtenir un primer prototip i han anat desenvolupant-se paral·lelament amb l'evolució de les aplicacions. Un cop es va obtenir una versió estable de les dues aplicacions es va iniciar una fase de proves més completa (tenint en compte que el desenvolupament de les dues aplicacions es va realitzar per separat, en moments diferents). Aquesta fase de proves es va basar en les metodologies de prova de Caixa Negra i Caixa Blanca.

7.3.1 MÈTODE CAIXA NEGRA

El mètode de prova de Caixa Negra²⁸ es centra en els resultats obtinguts segons els paràmetres introduïts prèviament per l'usuari, això és degut a que aquest no li interessen els detalls de funcionament del programari, tan sols la validesa dels resultats obtinguts a partir dels paràmetres d'entrada. És per això que es tracta el programari com una caixa tancada de la qual no interessa el tractament que realitza de les dades, si no el seu resultat final i d'aquí el nom del test. De totes formes perquè les proves siguin vàlides es generen una sèrie de dades d'entrada que compleixin les especificacions del problema. Els diferents tipus de dades que s'han de tractar són:

- Valors fàcils: Aquests també s'han de provar perquè és possible que si el programador es centra en els valors complexos, en ocasions els senzills no es compliran.
- Valors típics realistes: S'han de realitzar proves amb dades fàcilment comprovables manualment.
- Valors extrems: Molts programes cometen errors en els seus extrems, ja que és fàcil que els algorismes que contenen comptadors de cicle o fan referència a dimensions continguin errors d'una unitat.
- Valors il·legals: Quan s'introdueixen valors no possibles o de tipus diferent als tractats el resultat ha de ser un missatge d'error.

²⁸ (Díaz, J.F. 2007)

El mètode de Caixa Negra es centra en els requisits fonamentals del software i permet obtenir entrades que proven els requisits funcionals del programa. Amb aquest tipus de proves s'intenten trobar: funcions incorrectes o absents, errors d'interfície, errors en estructures de dades o en accés a les bases de dades externes, errors de rendiment i errors d'inicialització i finalització. Amb l'aplicació d'aquesta tècnica s'obté un conjunt de proves que redueixen el nombre de casos de prova i ens diuen alguna cosa sobre la presència o absència d'errors.

7.3.2 MÈTODE CAIXA BLANCA

El mètode de Caixa Blanca²⁹ parteix de la idea que un programari no ha estat completament testat si el seu codi conté parts que no han sigut executades mai. En aquest mètode s'analitza l'estructura lògica del programa i les dades idònies que conduiran a cada possible alternativa que es pugui presentar. Així doncs es procura escollir les dades que verifiquin cada possibilitat en les estructures de selecció múltiple, com per exemple en cada proposició if, cada alternativa en un switch i la condició de finalització de cada cicle.

En aquest mètode es prova cada mòdul per separat ja que el seu tamany facilita els tests i constitueix un excel·lent mitjà de proves i depuració. També acostuma a ser el més eficient per a descobrir errors, probablement perquè els errors més difícils d'identificar es troben en les interfícies dels subprogrames, degut a que no es comprenen correctament les condicions i normes precises d'intercanvi d'informació entre els subprogrames per part del/s programador/s. Per tant un bon criteri de prova per a grans projectes és aplicar el mètode de Caixa Blanca a cada petit mòdul conforme es vagi escrivint i posteriorment provar aquestes dades en tests de verificació més amplis.

7.3.3 PROVES IPHONE

Les primeres proves es van realitzar amb l'entorn destinat a aquest ús subministrat pel mateix SDK d'Apple, el iPhone Simulator. Paral·lelament al desenvolupament es van fer les proves de Caixa Blanca, sobretot es va utilitzar aquest mètode per a provar la descàrrega de la BBDD en el terminal mitjançant la connexió a un servidor de fitxers, però també quan es pitja una de les cel·les per a comprovar que les consultes a la BBDD eren les correctes i donaven els resultats adequats. Un cop finalitzada aquesta fase es van començar les proves en el terminal. Per a realitzar les proves en el terminal i poder transferir-hi la aplicació es va haver de subscriure una llicència de desenvolupament per part d'Apple (fins al moment s'havia evolucionat amb la subscripció gratuïta). Concretament es va subscriure la llicència anual "Standard Program", aquesta llicència està enfocada per a desenvolupadors que vulguin crear programari gratuït o de pagament per a posteriorment distribuir-lo a través de la App Store. Es va optar per aquesta llicència per a la generació del pilot, ja que el procés d'obtenció és més ràpid i el preu de la mateixa inferior. Un cop consolidat el projecte dins de la companyia l'objectiu és migrar a Enterprise, ja que Standard limita la distribució de l'aplicació a 100 terminals, degut a que l'Ad Hoc és un sistema de distribució per a la fase de proves i desenvolupament de programari.

²⁹ (Díaz, J.F. 2007)

En un primer cas es va optar per realitzar les proves amb el terminal connectat al portàtil mitjançant un cable USB i executant-lo a través del Xcode, així es pot seguir depurant el codi pas per pas mentre es prova. Però per a realitzar aquestes proves és necessari fer una sèrie d'accions:

- Entrar a l'apartat "Team" del lloc web iPhone Developer Program i afegir els diferents usuaris que formen part de l'equip de desenvolupament, assignant-li els diferents rols.
- Generar una signatura digital a través del Keychain Access, programari subministrat pel Mac OS X Leopard.
- Entrar a l'apartat Certificates del lloc web i pujar-hi la signatura digital.
- Posteriorment aprovar aquesta signatura en el mateix lloc web.
- Descarregar del lloc web i instal·lar aquesta signatura en el Mac.
- Guardar la clau privada interna a la signatura per a poder-la enviar a d'altres equips.
- Localitzar l'identificador únic del terminal a través del Xcode o del iTunes.
- Afegir els terminals a l'apartat "Devices" del lloc web.
- Generar un identificador de l'aplicació a l'apartat "App IDs" del lloc web.
- Crear un perfil de distribució per a desenvolupament a l'apartat "Provisioning" del lloc web.
- Instal·lar el perfil al terminal iPhone.
- Realitzar una sèrie de canvis al fitxer "info" del projecte.

Un cop finalitzades aquestes accions es realitzen les proves en el terminal. Es centren sobretot en l'obtenció de la BBDD a través de la connexió al servidor de fitxers, que aquesta BBDD s'obtingui correctament i en cas contrari es mostrin per pantalla els diferents errors que es puguin produir, tant si no hi ha connexió a la xarxa com si no es pot descarregar el fitxer. En el cas de que la connexió sigui correcta i s'obtingui el fitxer es mostra un missatge que indica la correcta descàrrega.

Per a continuar la realització de tests i amb el desenvolupament pràcticament finalitzat es va decidir fer extensible la fase de proves a més usuaris per a poder conèixer les seves impressions. Això implica avançar en el tema de la distribució, optant en aquest cas per a la distribució Ad Hoc. Degut a no disposar d'una llicència Enterprise no s'inclou el "In-house Distribution" (distribució interna a la companyia). La distribució iPhone facilita les coses podent subministrar el programari sense disposar físicament del terminal, només facilitant dos fitxers. Però per a poder realitzar això és necessari fer una sèrie de passos:

- Generar una nova signatura digital a través del Keychain Access, programari subministrat pel Mac OS X Leopard. Però en aquest cas per a la distribució.
- Entrar a l'apartat Certificates del lloc web i pujar-hi la signatura digital.
- Descarregar del lloc web i instal·lar aquesta signatura en el Mac.
- Guardar la clau privada interna a la signatura per a poder-la enviar a d'altres equips.
- Crear un nou perfil de distribució per a desenvolupament a l'apartat "Provisioning" del lloc web.
- Crear un nou perfil de configuració al projecte i realitzar una sèrie de canvis al fitxer info del projecte.
- Crear un nou fitxer al Xcode pel projecte del tipus "Code Signing" → "Entitlements".
- Verificar el correcte funcionament del programari.
- Preparar i inserir una icona amb els requeriments especificats al projecte (aquest pas no és necessari però si recomanable).

Un cop finalitzats aquests passos es pot instal·lar la aplicació en qualsevol iPhone, prèvia creació del terminal al lloc web en l'apartat "Devices". Per a poder instal·lar el programari és necessari disposar d'un ordinador, un iPhone, el seu corresponent cable USB i el programari iTunes d'Apple. L'aplicació i el perfil d'usuari són subministrats a l'usuari i aquest només ha de sincronitzar aquests fitxers amb l'iPhone mitjançant iTunes amb el terminal connectat a l'ordinador. Això està descrit amb tots els seus detalls en el manual d'ús del programari.

Amb l'aplicació instal·lada en el terminal es van repetir les proves de Caixa Blanca ja realitzades sobre l'iPhone Simulator obtenint els resultats desitjats. I amb els usuaris utilitzats per a realitzar les diferents proves d'ús de la aplicació es va aprofitar per a realitzar les proves de Caixa Negra, proves que es van superar satisfactòriament ja que els resultats obtinguts eren els esperats per les dades introduïdes.

7.2.4 PROVES APLICACIÓ D'ACTUALITZACIÓ

Paral·lelament al desenvolupament de l'aplicació en l'ordinador de sobretaula es realitzen les proves de Caixa de Blanca, on es testegen els diferents mòduls del programari, introduint-hi tot tipus de dades per aconseguir executar totes les parts del codi. D'aquesta forma es poden visualitzar les diferents reaccions del programari i tenir controlats tots els aspectes tractats. També s'aconsegueix depurar un munt d'errors que no es poden apreciar a simple vista o amb proves condicionades o subjectives, que resulten ser cabdals sobretot per evitar errors que pugui realitzar l'usuari per un mal ús del programari provocat pel desconeixement del mateix.

Amb aquest procés de creació i test finalitzat s'inicien les proves de Caixa Negra, en aquestes es testeja la idoneïtat dels resultats obtinguts partint d'una sèrie de valors prèviament introduïts. Per aquest tipus de prova es compta també amb la col·laboració d'alguns usuaris externs al desenvolupament per conèixer les seves opinions al respecte. Un cop els resultats són els desitjats s'evoluciona cap a la següent fase, la instal·lació de l'aplicació en el servidor i la seva posada en funcionament.

La aplicació s'instal·la en una màquina virtual VMWare ESX, allotjada en un servidor HP Proliant DL380 G5, la qual disposa d'un sistema operatiu Ubuntu. Per a la instal·lació d'aquesta aplicació és necessari instal·lar prèviament la versió 2.5 de Python (la utilitzada per Turbogears), el mateix Turbogears amb la versió utilitzada per al desenvolupament la 1.0.7, el motor de BBDD SQLite i les eines de treball SQL SQLAlchemy i SQLAlchemyObject. Un cop realitzats tots aquests passos és el moment d'exportar l'aplicació al servidor, fer els canvis indicats en la documentació de Turbogears (en aquest cas es realitza una instal·lació de tipus mod_wsgi, que és la recomanada per Turbogears) i provar el programari. El més normal en aquests casos és que no funcioni, ja que s'han de realitzar una sèrie de modificacions en el codi per a què el programari sigui capaç de trobar els fitxers tant de BBDD com d'altres amb els que treballa. Fets aquests canvis el funcionament de l'aplicació torna a ser el esperat, però es fan de nou les proves de Caixa Negra per a confirmar que el funcionament és el correcte. Finalitzades les proves de Caixa Negra la aplicació es posa en producció.

8. CONCLUSIONS

La realització d'aquest projecte aporta una nova eina de treball dins de la companyia, mitjançant la introducció d'una tecnologia innovadora en el si de l'empresa. La qual pot aportar un munt de solucions avançades per a facilitar i fer més còmode el treball dels seus empleats. També els introdueix en un nou sector de desenvolupament en ple creixement, el qual apunta a ser capdavanter en el món de la telefonia mòbil. Aquest desenvolupament es centra en el terminal d'Apple iPhone.

En el projecte també s'han intentat aprofitar les sinergies generades per el programari lliure i crear una aplicació web, interna a Solvay, per a poder actualitzar la BBDD associada a l'aplicació del telèfon. Aquesta aplicació ha estat realitzada amb l'objectiu que una persona amb uns coneixements mínims en l'ús d'ordinadors sigui capaç d'actualitzar els seus continguts. I permeti descarregar-la a d'altres usuaris des de qualsevol lloc on es disposi de connexió Wifi, 3G o GPRS.

S'ha conjugat també el coneixement i desenvolupament en una nova tecnologia amb la contribució amb un altre projecte molt important dins de la companyia, com és el Solvay Sustainable. Un projecte de llarg recorregut i de vital importància per la continuïtat i creixement futur de la companyia així com per el desenvolupament sostenible del nostre planeta. Aquesta conjugació permet incrementar la difusió del projecte Sustainable i col·laborar en la consecució dels seus objectius.

8.1 ACOMPLIMENT DELS OBJECTIUS

A continuació es repassaran els diferents objectius marcats en el inici, comentant l'estat de cadascun d'ells en el moment de la finalització del projecte:

- Conèixer suficientment l'entorn MAC OS per a poder treballar-hi amb la comoditat i rapidesa necessàries per a desenvolupar-hi:
 - Aquest objectiu ha sigut assolit satisfactòriament, aconseguint un nivell notable d'interacció amb el programari proporcionat per Apple.
- Descobrir les eines de desenvolupament en entorn MAC i iPhone, conèixer les seves particularitats i les diferents prestacions que proporcionen al desenvolupador:
 - Aquest objectiu ha sigut assolit satisfactòriament, coneixent i podent treballar de forma eficient amb totes les eines subministrades per al desenvolupament de programari. Com Xcode, Interface Builder, iPhone Simulator, etc...
- Aprendre el llenguatge de programació Objective-C: conèixer en profunditat les seves principals característiques per a poder desenvolupar aplicacions de forma ràpida i eficient:
 - Aquest objectiu ha sigut assolit satisfactòriament, tot i que degut als requeriments del projecte s'ha desenvolupat només la part associada al terminal iPhone. En alguns casos existeixen moltes similituds en la realització de programari per iPhone com per a sistemes Mac OS, però en la majoria de situacions existeixen diferències substancials

entre ambdós. Ja que es tracta del mateix llenguatge, per tant la sintaxi és la mateixa però canvien moltes llibreries, relacions entre objectes, etc...

- Analitzar les necessitats de l'empresa i com les solucions que aporta la tecnologia d'Apple ens poden ajudar a resoldre-les, sobretot en el cas de l'iPhone i les seves característiques:
 - Aquest objectiu també s'ha assolit, s'han analitzat les diferents solucions i funcionalitats que ofereix tant el terminal iPhone com el seu programari associat, destriant les possibilitats que poden ser útils per al desenvolupament de la nostra aplicació.
- Dissenyar un prototip d'aplicació utilitzant els coneixements adquirits i tenint en compte els requeriments de la companyia:
 - Aquest objectiu s'ha assolit satisfactòriament, això ho demostra la creació de la aplicació iSustainable.
- Enriquir aquest prototip amb múltiples millores successives que aportin noves solucions, enriqueixin la aplicació o en facilitin el seu ús:
 - Aquest objectiu s'ha assolit satisfactòriament, tot i que encara s'hi poden introduir noves millores, el desenvolupament d'aquesta aplicació ha arribat a una fase suficientment avançada com per a considerar-la operativa. S'hi han introduït diverses millores progressivament. Destacar sobretot l'apartat d'actualització de la BBDD i la millora de la presentació respecte les primeres versions més senzilles.
- Desenvolupar altres aplicacions paral·leles per a cobrir necessitats menys rellevants en el context de l'empresa:
 - Aquest objectiu no s'ha assolit però es tractarà més endavant en l'apartat de "Línies Obertes" d'aquest mateix capítol.
- Establir la tecnologia i els procediments necessaris per a futurs desenvolupaments en aquesta plataforma en el si de la companyia, els qual puguin prendre aquest desenvolupament com un punt de partida cap a noves aplicacions Apple o iPhone:
 - Aquest objectiu s'ha assolit satisfactòriament, ja que s'han posat les bases per a futurs desenvolupaments en tecnologia iPhone en el si de la companyia.

8.2 ALTRES ASPECTES A DESTACAR

Aquest projecte es pot considerar original o creatiu en si mateix, degut a que es tracta d'una nova tecnologia per la qual s'ha hagut d'aprendre i desenvolupar des de zero. Les seves aplicacions futures i les seves possibilitats són molt grans, tot just s'està començant el procés creatiu i en un futur no gaire llunyà es podrà apreciar millor l'evolució d'aquesta tecnologia. Com a aspectes negatius també s'ha destacar la necessitat de realitzar aplicacions internes al terminal per ha estar sempre disponibles, quan la informàtica cada cop més s'encamina cap a aplicacions web.

8.3 LÍNIES OBERTES

El treball realitzat ha obert una nova via de desenvolupament on seguir treballant i explorant altres solucions i aplicacions encaminades a millorar i ampliar els serveis que ofereix la SIS als altres departaments de Solvay en l'actualitat. Tant a nivell d'iPhone com a nivell de programari associat, una de les coses que es volen aconseguir es introduir l'iPhone en el catàleg de terminals mòbils de la companyia i que l'usuari rebi el terminal configurat amb el seu propi perfil, sincronització amb el compte de correu electrònic, etc... També es pot millorar la mateixa aplicació iPhone generada, introduint actualitzacions i millores a nivell de continguts, ampliant la informació mostrada amb més dades referents a cada centre o millorant la navegació entre els diferents centres i/o les diferents diapositives mitjançant els esdeveniments preconfigurats de les llibreries d'Objective-C.

També es volen generar altres aplicacions derivades del codi i arquitectura dissenyats per iSustainable, que ajudin al dia a dia dels treballadors implicats, facilitant la seva tasca i incrementin el rendiment de la companyia. Algunes d'aquestes aplicacions, en les que ja s'ha pensat són:

- **Balanced Scoreboard**, aplicació encaminada a mostrar els diferents valors de vendes, producció, costos, etc... Fins aquell moment dins dels últims intervals de temps: mesos, trimestres, anys, etc... Mitjançant un sistema de taules i pantalles amb la informació més important i de forma molt visual, on l'usuari pugui consultar la màxima informació en el mínim temps possible. Seria necessari que es realitzés una descàrrega de una BBDD per actualitzar la aplicació cada cop que s'iniciés l'aplicació. Aquest programari aniria destinat sobretot a personal directiu, també de vendes o d'altres departaments de compatibilitat i/o gestió.
- **Visualització de planells**, programari a realitzar per a emmagatzemar i mostrar els planells de les diferents plantes de producció de la companyia. Aquest programari està encaminat a facilitar la feina dels enginyers de planta, els quals han de desplaçar-se per l'interior de la fàbriques amb nombrosos tubs de planells. Amb aquesta aplicació i amb l'eina per a poder ampliar les imatges es podria reduir significativament la quantitat de plànols impresos, col·laborant d'aquesta manera amb la preservació del medi ambient. Una alternativa a aquest projecte amb iPhone seria la realització del mateix amb "Tablet PCs" però això queda fora de l'abast d'aquesta memòria.

El nombre de possibles aplicacions a realitzar és molt alt, però s'ha de tenir en compte una cosa molt important. En aquests moments s'han de crear aplicacions específiques en el cas de que hagin d'estar sempre disponibles, ja que no es disposa de una connexió econòmica a Internet permanent i en qualsevol localització. En el moment que aquesta situació canviï les aplicacions es realitzaran com a pàgines web adaptades al format iPhone, i seran accessibles des de qualsevol terminal sense necessitat d'instal·lació. En aquest sentit ja existeixen entitats que han apostat per aquest sistema, com per exemple el servei de venda d'entrades "Servi Caixa" de la Caixa d'Estalvis i Pensions de Barcelona.

8.4 VALORACIÓ PERSONAL I AGRAÏMENTS

Personalment la realització d'aquest projecte m'ha permès créixer a nivell professional, educatiu i humà. Professionalment he ampliat el meu currículum i la meva experiència laboral col·laborant en projectes importants dins d'una gran empresa, obrint-me nous camins professionals i ampliant els horitzons del meu futur. Educativament he progressat en diversos àmbits que desconeixia per complet i he ampliat els meus coneixements en d'altres; he aprofundit en l'aprenentatge de llenguatges de programació orientada a objecte com el Python o l'Objective-C, l'estructura i la programació de BBDD de tipus SQL o he descobert molts aspectes sobre les tecnologies Apple, les plataformes de generació de pàgines web com Turbogears o el programari lliure. Com a persona m'ha servit per créixer, familiaritzar-me en l'entorn de treball, aprofundir en el funcionament intern d'una gran empresa i conèixer noves persones capaces d'aportar-te coneixements o virtuts.

A més a més des d'aquestes línies voldria agrair la oportunitat que m'ha brindat Solvay per a realitzar aquest projecte en el si d'una gran companyia i especialment al departament SIS i els seus integrants per haver confiat en mi. També agrair a totes les persones que amb els seus comentaris o apreciacions han aportat el seu granet de sorra en la consecució dels objectius i la finalització amb èxit d'aquest projecte. A tots ells moltes gràcies.

9. BIBLIOGRAFÍA

1. Solvay SA. (Gener, 2009). *Solvay Ibèrica*. Consultat el Març de 2009, a <http://www.solvayiberica.es/>
2. Solina, E. (1999). *Libro Blanco '99*. Barcelona: Solvay SA.
3. Apple Team. (Desembre, 2008). *developer.apple.com*. Consultat el Abril de 2009, a <http://developer.apple.com/iphone/library/documentation/Cocoa/Conceptual/ObjectiveC/Introduction/introObjectiveC.html>
4. Anònim. (Desembre, 2008). *Wikipedia.com*. Consultat el Març de 2009, a <http://en.wikipedia.org/wiki/Objective-C>
5. Apple Team. (Desembre, 2008). *Xcode*. Consultat el Abril de 2009, a <http://developer.apple.com/TOOLS/xcode/>
6. Anònim. (Desembre, 2008). *Wikipedia.org*. Consultat l'Abril de 2009, a <http://en.wikipedia.org/wiki/Xcode>
7. Apple Team. (Desembre, 2008). *Interface Builder*. Consultat el Abril de 2009, a <http://developer.apple.com/tools/interfacebuilder.html>
8. Hipp, R. (Desembre, 2008). *About SQLite*. Consultat el Març de 2009, a <http://www.sqlite.org/about.html>
9. Hipp, R. (Desembre 2008). *SQLite Documentation*. Consultat el Març de 2009, a <http://www.sqlite.org/docs.html>
10. Anònim. (Desembre, 2008). *Wikipedia.org*. Consultat el Març / 2009, a <http://en.wikipedia.org/wiki/SQLite>
11. Dangoor, K. (Desembre, 2008). *Turbogears.com*. Consultat el Desembre de 2008, a <http://docs.turbogears.org/1.0>
12. Dangoor, K. (Desembre, 2008). *Turbogears.com*. Consultat el Desembre de 2008, a <http://turbogears.org/about/>
13. Anònim. (Març, 2009). *Wikipedia.org*. Consultat el Març de 2009, a <http://es.wikipedia.org/wiki/Widgets>
14. Anònim. (Desembre, 2008). *Wikipedia.com*. Consultat l'Abril de 2009, a <http://en.wikipedia.org/wiki/Model-view-controller>

15. Anònim. (Desembre, 2008). *Wikipedia.com*. Consultat el Novembre / 2008, a <http://en.wikipedia.org/wiki/MochiKit>
16. MochiMedia. (Desembre, 2008). *Mochikit.com*. Consultat el Gener de 2009, a <http://mochikit.com/about.html>
17. Robert. (Juliol, 2007). *Kid-templating.org*. Consultat el Desembre de 2008, a <http://www.kid-templating.org/>
18. Anònim. (Desembre, 2008). *Wikipedia.com*. Consultat el Gener de 2009, a <http://en.wikipedia.org/wiki/CherryPy>
19. Edgewall Software Team. (Desembre, 2008). *CherryPy.org*. Consultat l'Abril de 2009, a <http://www.cherrypy.org/>
20. Python Software Foundation. (Desembre, 2008). *Python.org*. Consultat l'Abril de 2009, a <http://www.python.org/>
21. Python Software Foundation. (2008, Desembre, 2008). *Python.org About*. Consultat l'Abril de 2009, a <http://www.python.org/about/>
22. Python Software Foundation. (Desembre, 2008). *Python.org Documentation*. Consultat l'Abril de 2009, a <http://www.python.org/doc/>
23. Anònim. (Desembre, 2008). *Wikipedia.org*. Consultat l'Abril de 2009, a <http://es.wikipedia.org/wiki/Python>
24. SQLAlchemy Team. (Desembre, 2008). *The Python SQL Toolkit and Object Relational Mapper*. Consultat el Febrer de 2009, a <http://www.sqlalchemy.org/>
25. SQLAlchemy Team. (Desembre, 2008). *SQLAlchemy 0.5.3 Documentation*. Consultat el Febrer de 2009, a <http://www.sqlalchemy.org/docs/05/index.html>
26. SQLAlchemy Team. (Desembre, 2008). *Key Features of SQLAlchemy*. Consultat el Febrer de 2009, a <http://www.sqlalchemy.org/features.html>
27. Anònim. (Desembre, 2008). *Wikipedia.org*. Consultat el Març de 2009, a <http://en.wikipedia.org/wiki/SQLAlchemy>
28. Díaz, J. (2007). *Métodos de Prueba de Programas*. Consultat el Maig de 2009, a http://www.galeon.com/neoprogramadores/met_test.htm
29. Díaz, J. (2007). *Métodos de Prueba de Programas*. Consultat el Maig de 2009, a http://www.galeon.com/neoprogramadores/met_test.htm

ANEX 1: MANUAL D'USUARI ISUSTAINABLE

1. REQUISITS MÍNIMS

- Terminal iPhone.
- PC, Mac o portàtil amb connexió a Internet.
- Programari iTunes d'Apple.
- Cable de connexió USB iPhone-PC d'Apple (figura nº 1).



Figura nº 1

2. INSTAL·LACIÓ

Per a la instal·lació del programari en el terminal iPhone s'utilitza el sistema de distribució Ad-hoc d'Apple per a la distribució d'aplicacions. El sistema d'instal·lació el dividim en diferents passos que detallem a continuació:

PAS 1: CONTACTAR AMB L'ADMINISTRADOR

Per a començar la instal·lació és necessari subministrar el "Device ID" (40 caràcters hexadecimal) a l'administrador de l'aplicació. Per aconseguir aquest identificador hem de tenir el terminal connectat al PC mitjançant el cable USB i el iTunes arrancat.

Com podem observar a la figura nº 2, la qual és la pantalla principal del iTunes, existeix un camp "Identifier". És necessari per a la visualització d'aquest camp tenir seleccionat en el menú de l'esquerra en la pestanya "Devices" del nostre terminal iPhone.

És possible que aquest número d'identificador no aparegui directament en aquesta pantalla, en cas contrari s'ha de pitjar damunt del camp "Serial Number" i apareixerà seguidament.



Figura nº 2

PAS 2: DESCARREGAR L'APLICACIÓ

Una vegada hem rebut l'autorització de l'administrador per a instal·lar-la és necessari connectar-me a l'adreça web isustainable.sisdesk.com. I pitjar sobre "Download zip file" i posteriorment "Download profile" per ha obtenir els arxius necessaris per a la seva posterior instal·lació al terminal.

PAS 3: INSTAL·LACIÓ

Usuaris Windows: seleccionar ambdós fitxers descarregats, arrossegar-los fins a pestanya "Library" del menú de l'esquerra del iTunes i finalment sincronitzar l'iPhone amb l'iTunes.

Usuaris Mac OS: seleccionar ambdós fitxers i arrossegar-los fins a la icona del programari iTunes. Posteriorment sincronitzar l'iPhone amb l'iTunes.

En ambdós casos és necessari tenir arrencada l'aplicació iTunes i el terminal iPhone connectat al PC mitjançant el cable USB.

Un cop realitzades aquestes accions l'aplicació està instal·lada al dispositiu i preparada per al seu funcionament. Comprovar que apareix la icona del projecte Solvay Sustainable en algun dels menús principals de l'iPhone.

3. FUNCIONAMENT

Per arrancar l'aplicació és necessari pitjar sobre la icona "iSustainable" al menú principal del terminal. Al principi ens apareix la pantalla "SITES" amb una taula amb els diferents centres, els quals es poden seleccionar per a accedir al contingut de cadascun o es pot activar el menú d'actualització, local·litzat en el botó de la barra superior. Com es pot observar a la figura nº 3:

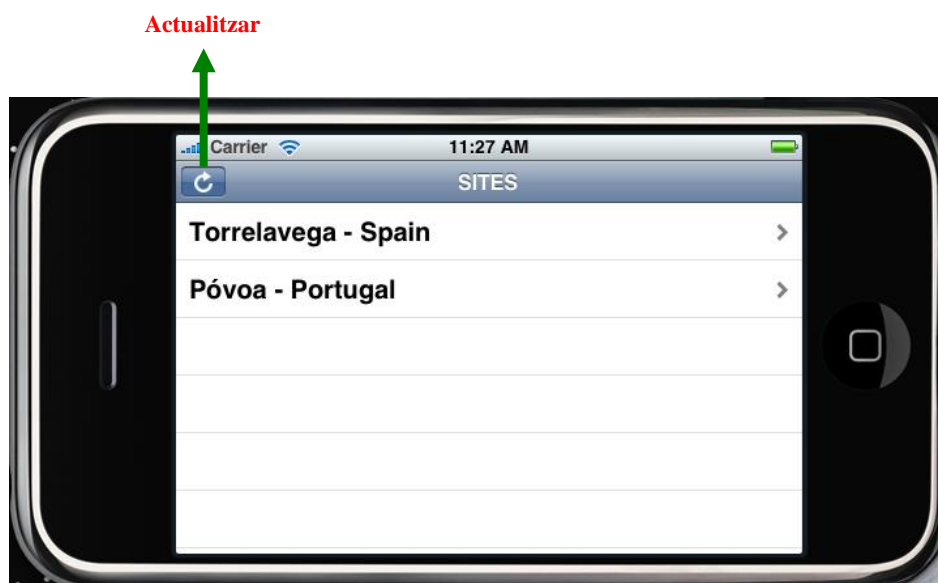


Figura nº 3

En pitjar damunt d'un dels centres canviem de pantalla i accedim a la pantalla índex per aquest centre. Aquesta nova pantalla és la matriu 5x5 del projecte Solvay Sustainable, com es pot observar en la figura nº 4. Per accedir al contingut generat em de pitjar directament damunt de cada cel·la, també es pot tornar enrere al menú de selecció de centres.

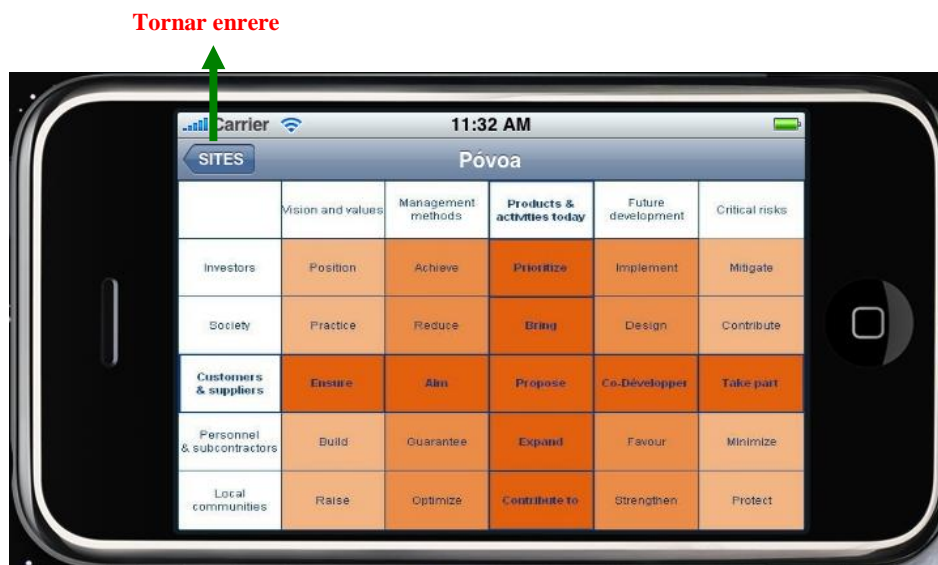


Figura nº 4

Després de pitjar damunt d'una de les cel·les canviem de pantalla i accedim a la pantalla "Details" de l'aplicació, figura nº 5, en aquesta visualitzem el contingut de cada cel·la. Aquest contingut es pot ampliar o reduir a voluntat dintre d'uns límits raonables, això és possible pitjant en dues posicions diferents de la pantalla i realitzant un moviment cap a l'exterior per a ampliar i cap al interior per a reduir. Realitzant doble pitjar sobre la imatge, aquesta recupera el seu tamany inicial.

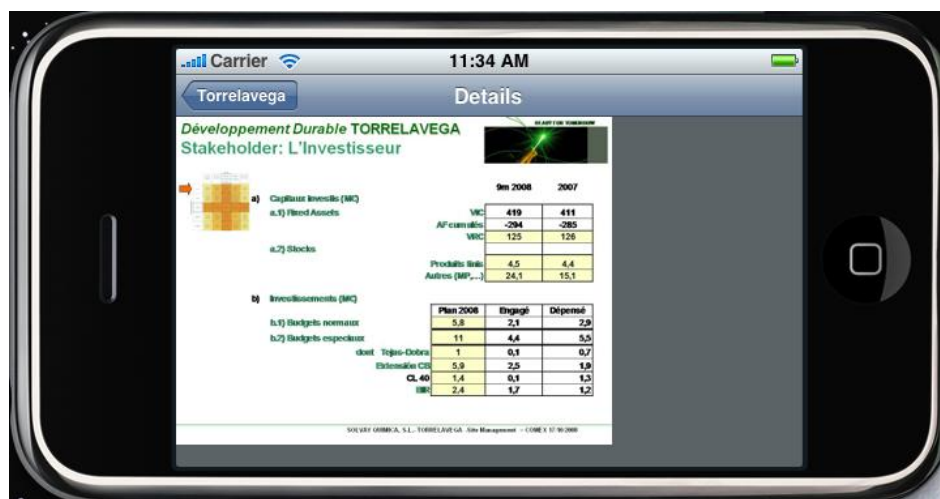


Figura nº 5

Tornant a la figura nº 3 o "SITES", en pitjar el botó d'actualització s'accedeix a la pantalla per a descarregar noves dades per a l'aplicació. Aquesta pantalla disposa de dos botons "DOWNLOAD" per a iniciar la descàrrega i "CANCEL" per a cancel·lar l'acció i tornar al menú inicial. Com podem observar en la figura numero 6.



Figura nº 6

En el cas de pitjar el boto "DOWNLOAD" s'inicia l'actualització de l'aplicació, com podem observar en la figura número 7:



Figura nº 7

Un cop inicialitzada l'actualització, es poden produir tres possibles resultats:

- Base de dades descarregada correctament, amb la qual cosa és necessari pitjar "OK", sortir de l'aplicació i tornar-la a arrencar, segons figura numero 8.

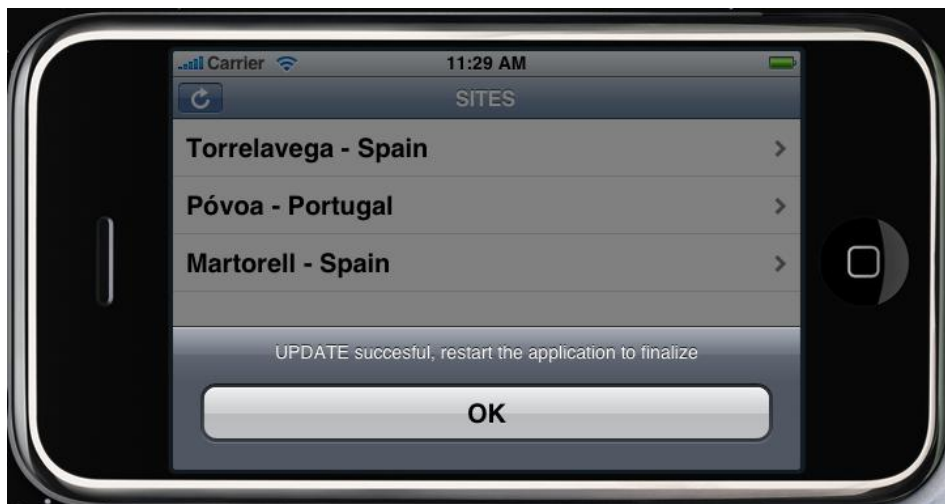


Figura nº 8

- S'ha produït un error degut a que el fitxer especificat no existeix o no està penjat al servidor o s'ha produït un problema amb el servidor, figura numero 9. En aquest cas consultar amb l'administrador.

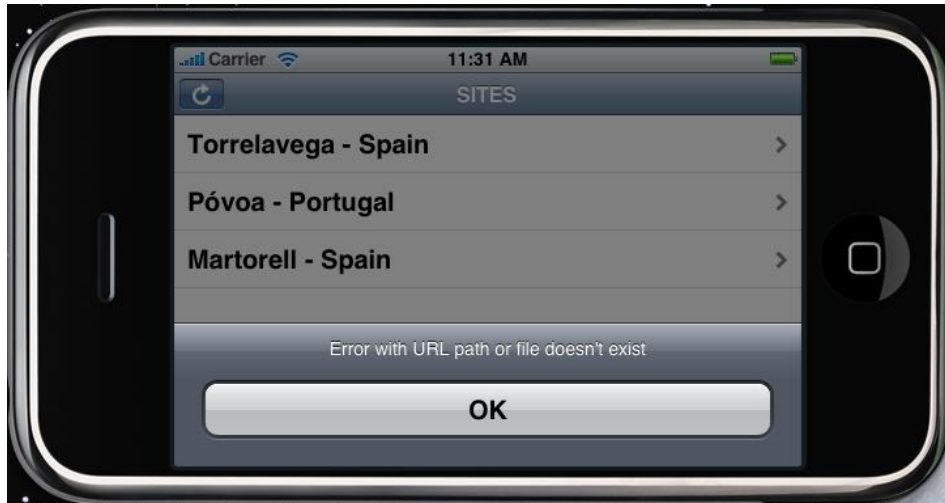


Figura nº 9

- El terminal no disposa de connexió a la xarxa, figura numero 10. Consultar l'estat de les connexions del terminal. En el cas d'estar totes connectades, existeix la possibilitat que no disposem de cobertura o que el terminal no funcioni correctament.

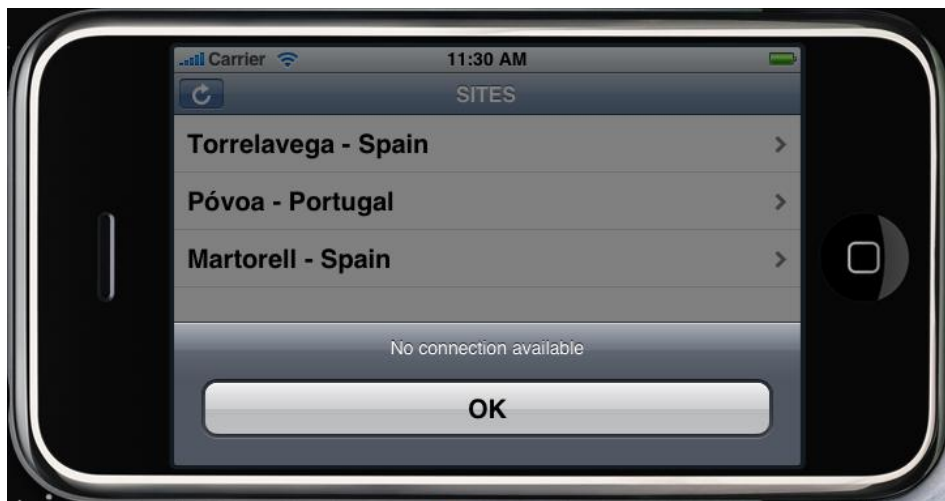


Figura nº 10.

També comentar que el funcionament és el mateix tan si tenim el terminal en qualsevol de les posicions horitzontals o verticals, tal i com podem comprovar a la figura numero 11.



Figura nº 11

ANEX 2: MANUAL D'USUARI APLICACIÓ D'ACTUALITZACIÓ

1. REQUISITS MÍNIMS

- Màquina virtual Servidor VMWare ESX.
- Servidor HP Proliant DL380 G5.
- Sistema Operatiu Ubuntu per a servidor.
- PC, Mac o portàtil connectat a la xarxa interna de Solvay.
- Navegador Internet: Internet Explorer, Mozilla Firefox, Google Chrome, Safari, etc...

2. INSTAL·LACIÓ

Per a la utilització d'aquest programari no és necessària la instal·lació en el terminal, ja que l'usuari és connecta a un servidor local mitjançant el navegador per defecte del Sistema Operatiu (SO) o algun altre que s'hagi instal·lat en el mateix.

3. FUNCIONAMENT

Per utilitzar aquesta aplicació és necessari obrir el navegador Internet del que disposa el SO (Internet Explorer) o algun altre que l'usuari es pot descarregar i instal·lar com Mozilla Firefox, Google Chrome o Safari (en el cas de terminal Mac). Tot i que l'aplicació ha estat desenvolupada i es recomana l'ús d'Internet Explorer o Mozilla Firefox. Un cop arrencat el navegador s'escriu la direcció: isustainable.sisdesk.com i et connecta amb l'aplicació Solvay Sustainable.

PAS 1: IDENTIFICACIÓ

Primerament ens apareix una pantalla d'identificació on l'usuari haurà d'identificar-se mitjançant un nom d'usuari i una contrasenya que seran proporcionats per l'administrador de l'aplicació, amb el qual hauran de posar-se en contacte. En la figura nº 1 apareix la pantalla d'identificació.

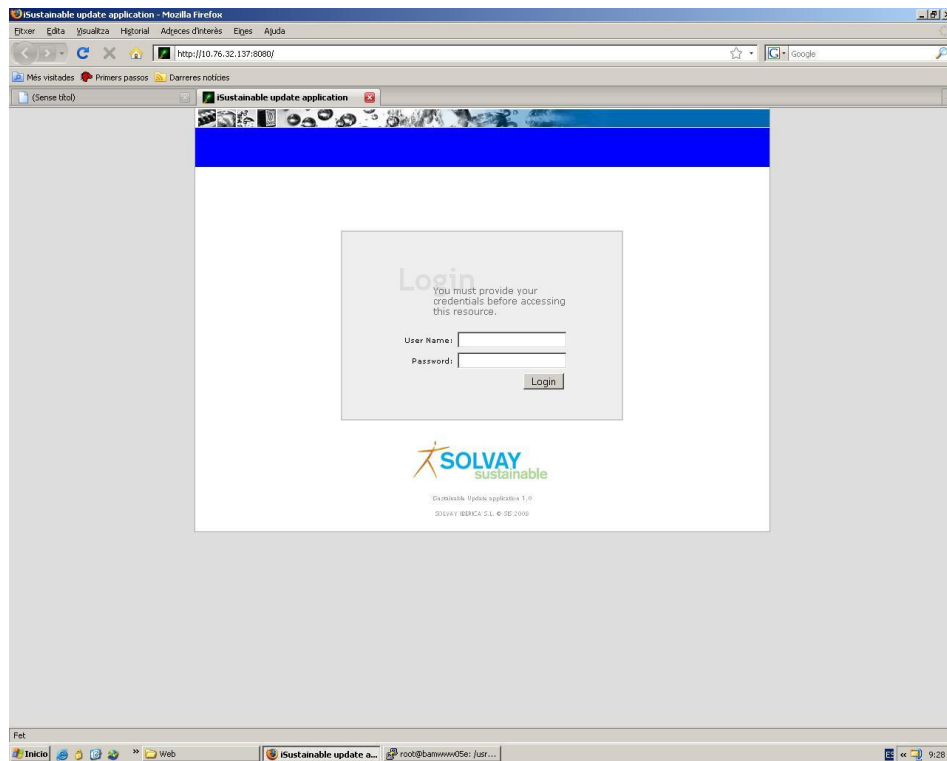


Figura nº 1.

PAS 2: MENÚ PRINCIPAL

Un cop introduït correctament nom d'usuari i contrasenya s'accedeix a la pantalla de la figura nº 2, el menú principal. En aquest menú es poden realitzar fins a cinc accions diferents representades mitjançant enllaços a d'altres pantalles. Aquestes accions són:

- Editar Base de Dades (BBDD).
- Importar BBDD.
- Descarregar fitxer *.zip amb l'aplicació iSustainable per iPhone.
- Descarregar perfil d'iPhone.

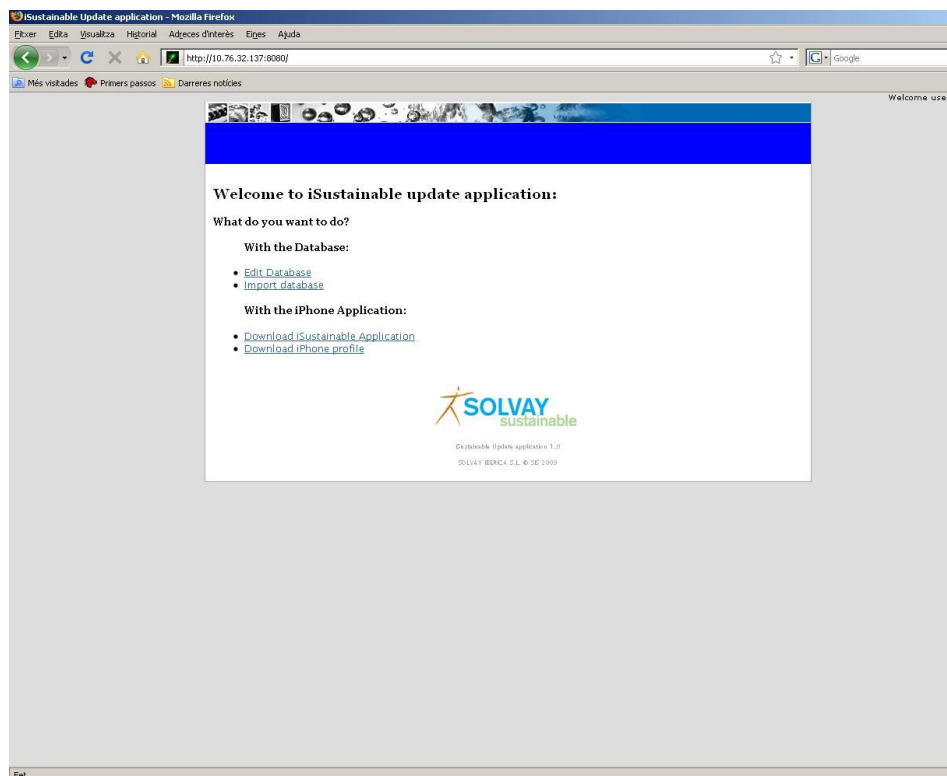


Figura nº 2.

PAS 3: EDIT DATABASE

Si es selecciona la primera opció “Edit Database” s’accedeix al mòdul principal de la aplicació, la qual permet treballar i realitzar canvis sobre la BBDD. En aquesta primera pantalla del mòdul (figura nº 3) es poden afegir centres, eliminar centres, seleccionar un centre per editar-lo o guardar els canvis realitzats.

- **Afegir Centre:** per afegir un centre nou és necessari col·locar-se en la casella “Center” i introduir el nom del centre, seguidament col·locar-se en la casella “Country” i introduir el país on es troba el centre i finalment pitjar en el botó “Add”. Un cop fet això el centre apareixerà en la llista de centres existents.
- **Eliminar Centre:** per eliminar un centre ja existent l’usuari ha de seleccionar-lo dins de la llista “Existing Centers” i posteriorment pitjar el botó “Delete”, s’ha de tenir cura en la utilització d’aquest botó, ja que elimina el centre seleccionat i tota la informació relacionada amb el centre.
- **Guardar els canvis:** per a guardar els canvis realitzats i publicar la BBDD en el servidor és necessari pitjar en el botó “Save changes”.
- **Editar centre:** per editar un centre ja existent és necessari seleccionar el centre de la llista “Existing centers” i pitjar el botó “Edit”, es passarà a una altra pantalla.

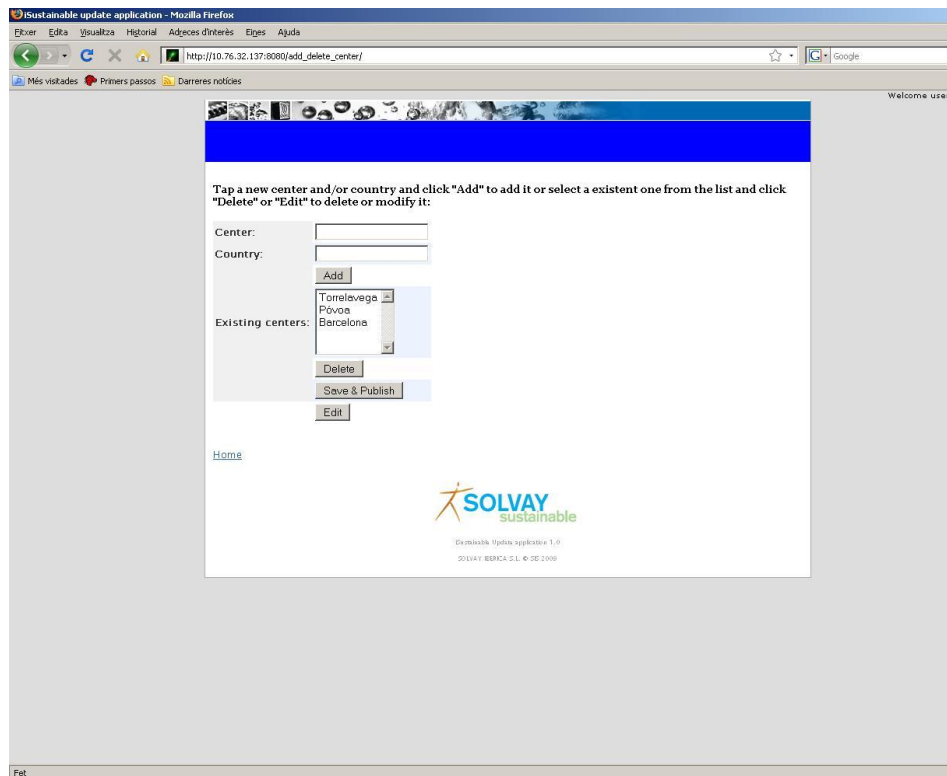


Figura nº 3.

PAS 4: SELECCIONAR CEL·LA DE LA MATRIU SOLVAY SUSTAINABLE

Un cop seleccionat el centre s'accedeix a una nova pantalla on es pot veure la matriu del projecte Solvay Sustainable, la mateixa que es visualitza en el terminal iPhone. Per a seleccionar la cel·la que es vol editar només s'ha de pitjar sobre de la mateixa i es redirecciona a una altra pantalla on modificar el contingut que es mostrarà a través de la cel·la. En aquesta pantalla també es mostra informació del centre seleccionat, mostrant el nom del centre i el país en la part superior de la pantalla. En la figura nº 4 es poden observar els detalls que s'han comentat.

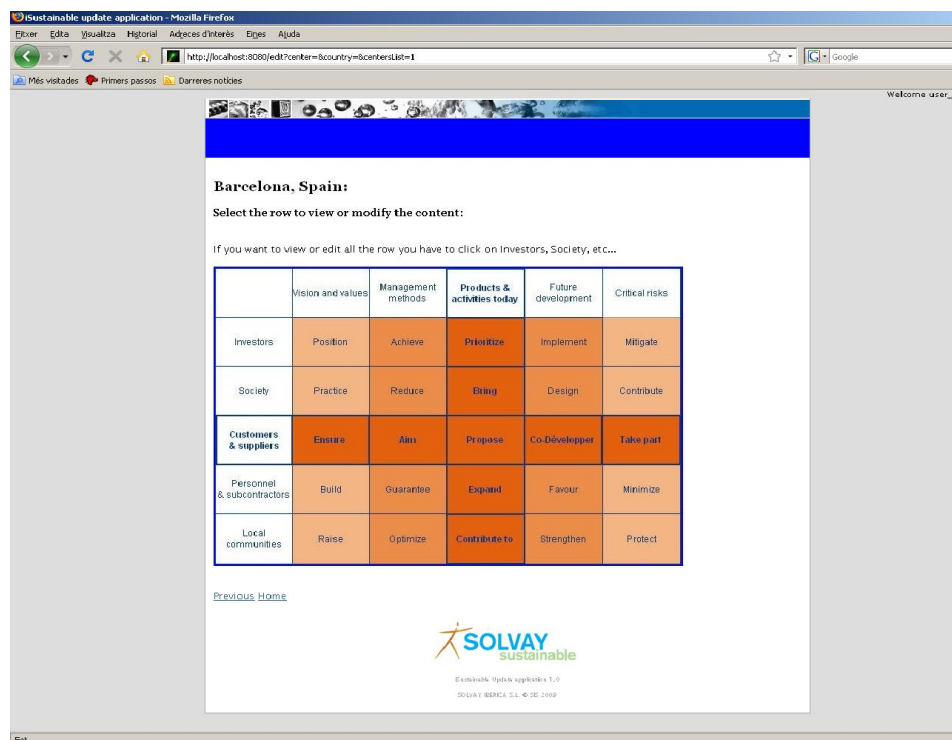


Figura nº 4.

PAS 5: MODIFICAR EL CONTINGUT DE LA CEL·LA

Quan ja s'ha seleccionat la cel·la, s'accedeix a la pantalla on es visualitza el contingut de la mateixa. En aquesta pantalla apareix en primer terme el contingut de la cel·la, o sigui la imatge que té relacionada. En cas de no tenir cap imatge relacionada apareix un espai en blanc. Sota de qualsevol de les dues coses hi ha el botó "Delete Image", el qual elimina la imatge introduïda en el cas de haver-hi alguna. Seguidament s'ha realitzat una caixa relacionada amb el botó "Navega..." (depenen segons l'idioma del navegador) per a introduir una nova imatge a la cel·la, la qual et permet seleccionar qualsevol fitxer que l'usuari disposi en les unitats locals i també una casella de text on introduir comentaris sobre la imatge o la cel·la. Per a finalitzar hi ha el botó "Tramet la consulta" (depenen de l'idioma del navegador) la funció del qual es guardar els canvis realitzats en la pantalla. En la figura nº 5 i nº 6 es poden veure les apreciacions que hem realitzat, amb o sense imatge introduïda respectivament.

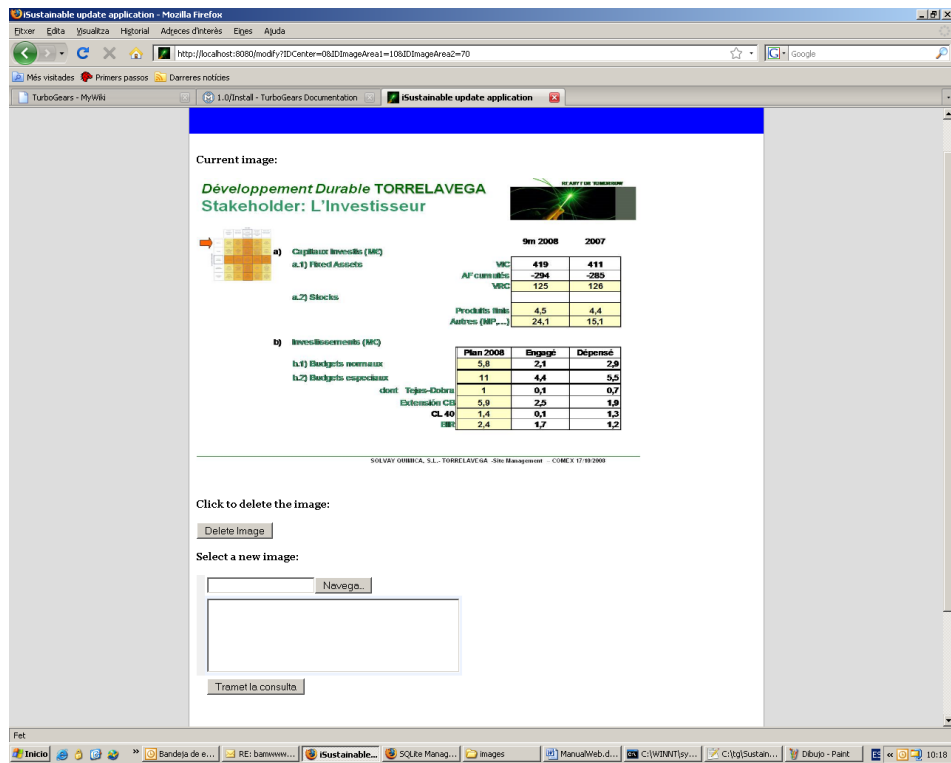


Figura nº 5.

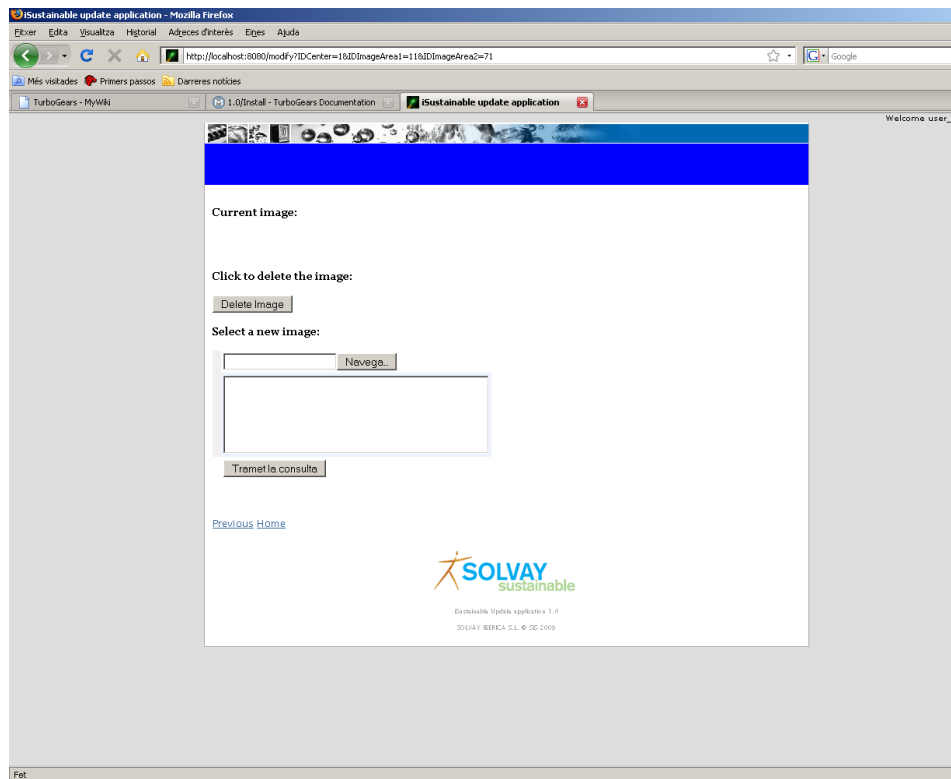


Figura nº 6.

PAS 6: IMPORT DATABASE, DOWNLOAD ZIP FILE I DOWNLOAD PROFILE

Si es selecciona alguna de les tres últimes opcions, s'obre un missatge del explorador per a descarregar-se la BBDD o el fitxer zip amb l'aplicació o el fitxer del perfil. S'ha de tenir seleccionada la opció "Desa el fitxer" i pitjar "D'acord" (depenen de l'idioma del navegador). El fitxer (en qualsevol dels tres casos) es quedarà emmagatzemat a l'escriptori de l'ordinador per defecte o l'ubicació que l'usuari triï. La única cosa que canviarà és el fitxer que es descarregarà. En la figura nº 8 es pot observar el que s'ha comentat. Per a conèixer com emprar els fitxers de perfil i de tipus zip consultar el Manual d'usuari d'iPhone.

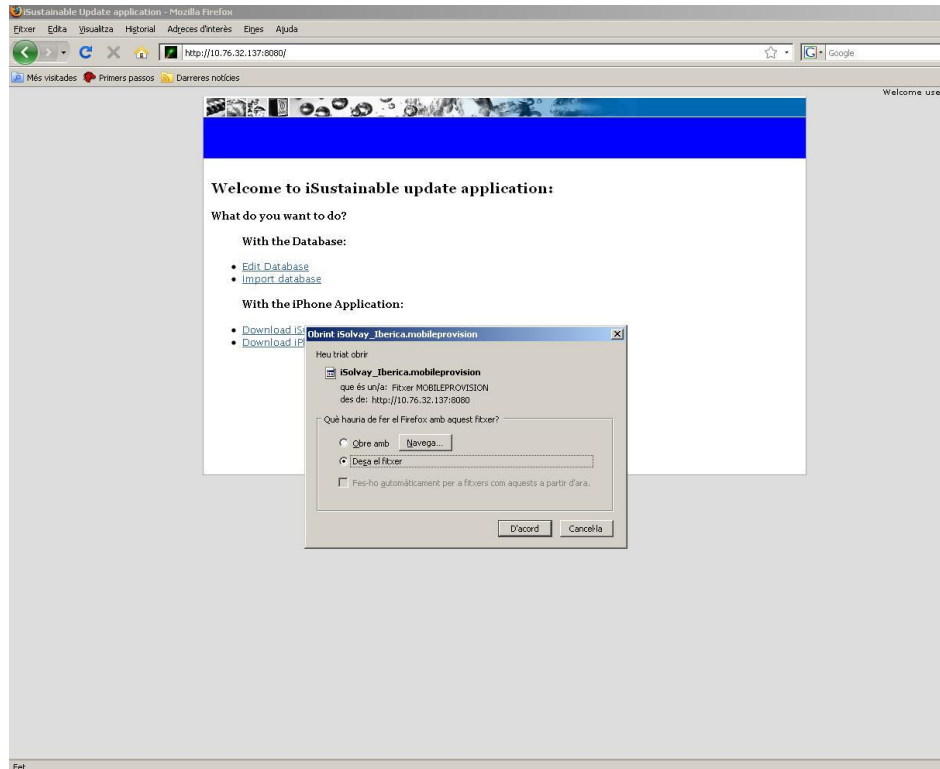


Figura nº 7.

